FGI PUBLICATIONS N:O 158

Context Awareness for Navigation Applications

BY Robert E. Guinness

NLS
FINNISH GEOSPATIAL RESEARCH INSTITUTE
FGI

FGI PUBLICATIONS № 158

Context Awareness for Navigation Applications

by

Robert E. Guinness

Doctoral dissertation for the degree of Doctor of Science in Technology to be presented with due permission for public examination and debate in Tietotalo Building, Auditorium TB109, at Tampere University of Technology on the 21st of December 2015 at 12 noon.

KIRKKONUMMI 2015

Supervising professor
Professor Jarmo Takala, Department of Pervasive Computing, Tampere University of Technology, Finland

Thesis advisors
Professor Ruizhi Chen, Conrad Blucher Institute for Surveying and Science, School of Engineering and Computer Science, Texas A&M University – Corpus Christi, USA

Professor Heidi Kuusniemi, Department of Navigation and Positioning, Finnish Geospatial Research Institute, National Land Survey of Finland

Preliminary examiners
Dr. Martin Werner, Mobile and Distributed Systems Group, Institute for Informatics, Ludwig-Maximilians-Universität München, Germany

D.Sc.(Tech) Susanna Pirttikangas, Department of Computer Science and Engineering, University of Oulu, Finland

Opponents
Dr. Martin Werner, Mobile and Distributed Systems Group, Institute for Informatics, Ludwig-Maximilians-Universität München, Germany

D.Sc.(Tech) Jari Syrjärinne, HERE, a Nokia business, Finland

ISBN (printed): 978-951-48-0249-2
ISBN (pdf): 978-951-48-0250-8
ISSN (print): 2342-7345
ISSN (online): 2342-7353

Grano Oy, Vantaa 2015

ABSTRACT

This thesis examines the topic of context awareness for navigation applications and asks the question, “What are the benefits and constraints of introducing context awareness in navigation?” Context awareness can be defined as a computer’s ability to understand the situation or context in which it is operating. In particular, we are interested in how context awareness can be used to understand the navigation needs of people using mobile computers, such as smartphones, but context awareness can also benefit other types of navigation users, such as maritime navigators. There are countless other potential applications of context awareness, but this thesis focuses on applications related to navigation. For example, if a smartphone-based navigation system can understand when a user is walking, driving a car, or riding a train, then it can adapt its navigation algorithms to improve positioning performance.

We argue that the primary set of tools available for generating context awareness is machine learning. Machine learning is, in fact, a collection of many different algorithms and techniques for developing “computer systems that automatically improve their performance through experience” [1]. This thesis examines systematically the ability of existing algorithms from machine learning to endow computing systems with context awareness. Specifically, we apply machine learning techniques to tackle three different tasks related to context awareness and having applications in the field of navigation: (1) to recognize the activity of a smartphone user in an indoor office environment, (2) to recognize the mode of motion that a smartphone user is undergoing outdoors, and (3) to determine the optimal path of a ship traveling through ice-covered waters. The diversity of these tasks was chosen intentionally to demonstrate the breadth of problems encompassed by the topic of context awareness.

During the course of studying context awareness, we adopted two conceptual “frameworks,” which we find useful for the purpose of solidifying the abstract concepts of context and context awareness. The first such framework is based strongly on the writings of a rhetorician from Hellenistic Greece, Hermagoras of Temnos, who defined seven elements of “circumstance”. We adopt these seven elements to describe contextual information. The second framework, which we dub the “context pyramid” describes the processing of raw sensor data into contextual information in terms of six different levels. At the top of the pyramid is “rich context”, where the information is expressed in prose, and the goal for the computer is to mimic the way that a human would describe a situation.

We are still a long way off from computers being able to match a human’s ability to understand and describe context, but this thesis improves the state-of-the-art in context awareness for navigation applications. For some particular tasks, machine learning has succeeded in outperforming humans, and in the future there are likely to be tasks in navigation where computers outperform humans. One example might be the route optimization task described above. This is an example of a task where many different types of information must be fused in non-obvious ways, and it may be that computer algorithms can find better routes through ice-covered waters than even well-trained human navigators. This thesis provides only preliminary evidence of this possibility, and future work is needed to further develop the techniques outlined here. The same can be said of the other two navigation-related tasks examined in this thesis.

PREFACE

The research work presented in this thesis was carried out between November 2011 and May 2014. I have chosen to complete a “compendium-style” dissertation, in part because I have already had the pleasure of preparing a monograph when co-authoring a book with Prof. Ruizhi Chen, published in July 2014. I have no great desire to repeat such an experience yet. As many who have published such monographs can attest, it takes a lot out of you!

Due to other responsibilities, as well as a bad case of the “it’s-not-good-enough-yet” syndrome, it took me more than one year to finalize and publish some of the results of my doctoral research in article format. With the aid of gentle nudging from my colleagues and superiors, I prepared the summary content for this compendium mostly between January and July 2015.

There are two particular experiences I’d like to share that also motivated me for completing this dissertation. The first is when I was asked to be a reviewer for an article submitted to one highly-esteemed journal. When I realized that my work, in my own opinion, was superior to that which I was reviewing, I felt suddenly cured of the above-mentioned syndrome. This is one of the side benefits of peer review. The second was when I was participating in an interview of a now colleague (he got the job!). We asked him if he could describe one achievement of which he was most proud. Instead of pointing to one particular academic achievement, such as a highly-cited paper, he pointed out another kind of achievement: the fact that he can look back at his publications and realize that some of the early ones were poor but that there has been a steady improvement in the quality over the years. Since hearing that, this is what I aim for: Not to publish the perfect gem some day but to continually put out my work-in-progress for others to see and hopefully benefit from. Then, refine and repeat.

There are many I would like to thank for their contributions to this thesis and to my overall development. First and foremost, however, I thank God for the wonderful life and opportunities He has given me and for always being with me. Next, I’d like to thank my family for their many years of support, beginning with my parents but also including my siblings, Erin and Joe. Nowadays I have my own family, and when someone completes a dissertation in the midst of family life, there is often an unsung hero (or heroine) behind the work. In my case, it is my wife, Anne-Mari, who spent many long days and evenings taking care of our boys while I was completing this thesis. Thank you, my angel, for everything. Then, to my children Pyry, Kilian, and Aarre: Thank you for bringing joy to each day.

Next, I'd like to thank my thesis supervisors, Prof. Jarmo Takala, Prof. Ruizhi Chen, and Prof. Heidi Kuusniemi. Thank you for all the support and guidance, as well as the opportunity to complete this research under your supervision. Thank you also to Prof. Ling Pei, who served as an instructor during my early days at FGI. Thank you to my pre-examiners, Dr. Martin Werner (who will also serve as an opponent at my defense) and Dr. Susanna Pirttikangas. A special thanks goes to Dr. Jari Syrjärinne who agreed to be my second opponent on short notice. I'd also like to thank Dr. Valérie Renaudin, who also agreed to be an opponent, but due to a last minute change in the defense date was unable to attend.

Lastly, I'd like to thank my colleagues and co-authors, who have made the environment in which this work was completed very fun, interesting, and rewarding. There are so many of you nowadays that I won't attempt to name everyone individually, but please know that I value each and every one of you. I have learned so much from my colleagues, and you have made it a joy to come to work each day.

Kirkkonummi, 20.11.2015

Robert E. Guinness

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ACM	Association for Computing Machinery
AdaBoost	Adaptive Boosting
ADL	Activities of Daily Life
AESS	Aerospace and Electronics Systems Society
AI	Artificial Intelligence
AIS	Automatic Identification System
ANN	Artificial Neural Network
API	Application Programming Interface
BC	Before Christ
BN	Bayesian Network
CHMM	Coupled Hidden Markov Model
COTS	commercial-off-the-shelf
CPU	Central Processing Unit
CRF	Conditional Random Fields
DBSCAN	density-based spatial clustering of applications with noise
DT	Decision Tree
DTS	discriminative temporal smoothing
E	East

EM	expectation-maximization
Five Ws	Who, What, Where, When, and Why
FMS	Future Urban Mobility Survey
GIS	Geographic Information Systems
GMM	Guassian mixture model
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPSAR	Google Play Services Activity Recognition
HCI	Human-Computer Interaction
HDA	Hazard Detection and Avoidance
HF-SVM	Hardware-friendly Support Vector Machines
HIER	hierarchical agglomerative clustering
HMM	Hidden Markov Models
IBk	instance-based k-nearest neighbor algorithm
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
ION	Institute of Navigation
IRO-2	Ice Forecast and Route Optimization
kMC	k-means clustering
kNN	k-Nearest Neighbor

KStar	K* algorithm
lat	latitude
LBS	Location-Based Services
LDA	Linear Discriminant Analysis
LoMoCo	Location-Motion-Context
long	longitude
LR	logistic regression
LS-SVM	Least Squares-Support Vector Machine
LSA	latent semantic analysis
LWL	Locally weighted learning
MAP	maximum a posteriori
MCMC	Markov chain Monte Carlo
MEMS	microelectromechanical systems
MLE	maximum likelihood estimate
MLP	Multilayer Perceptron
N	North
NB	Naïve Bayes
OPTICS	ordering points to identify the clustering structure
OS	Operating System
PCA	principal component analysis
PDA	Personal Digital Assistant
PLANS	Position Location and Navigation Symposium

RF	RandomForest
RSSI	Received Signal Strength Indicator
SAR	Synthetic Aperture Radar
SLAM	Simultanous Localization and Mapping
SVM	Support Vector Machines
UAVs	Unmanned Aerial Vehicles
UCI	University of California, Irvine
WGS84	World Geodetic System 1984
WLAN	wireless local area network

LIST OF PUBLICATIONS

This thesis contains a compilation of five previously published papers, referred to as [P#] throughout the text. The following publications are included:

[P1] R. Chen, **R. E. Guinness**, "Context Awareness" in *Geospatial computing in mobile devices*. Boston: Artech House, ch. 8, pp. 150-170, 2014.

[P2] R. Chen, **R. E. Guinness**, "Contextual Reasoning" in *Geospatial computing in mobile devices*. Boston: Artech House, ch. 9, pp. 171-197, 2014.

[P3] L. Pei, **R. E. Guinness**, R. Chen, J. Liu, H. Kuusniemi, Y. Chen, L. Chen, and J. Kaistinen, "Human behavior cognition using smartphone sensors," *Sensors*, vol. 13, no. 2, pp. 1402–1424, 2013.

[P4] **R. E. Guinness**, "Beyond where to how: A machine learning approach for sensing mobility contexts using smartphone sensors," *Sensors*, vol. 15, no. 5, pp. 9962–9985, 2015.

[P5] **R. E. Guinness**, J. Saarimäki, L. Ruotsalainen, H. Kuusniemi, F. Goerlandt, J. Montewka, R. Berglund, and V. Kotovirta, "A method for ice-aware maritime route optimization," in *Position, Location and Navigation Symposium–PLANS 2014*, IEEE/ION, pp. 1371–1378, 2014.

1. INTRODUCTION

1.1 Background and Motivation

We are currently witnessing an era of technological convergence that rivals some of the great technological upheavals of modern history[1]. The steam engine, the electric lamp, the transistor, the jetliner, the artificial satellite—it is in this same revered company that we can place the technological revolution we are now undergoing. According to authors Erik Brynjolfsson and Andrew McAfee, we are living in a "second machine age" (where the first machine age began with James Watt's steam engine), which they describe as "an inflection point in the history of our economies and societies because of digitization" [2, p. 11]. They define digitization as "converting things into bits that can be stored on a computer and sent over a network" [2, p. 10]. The resulting digital information has remarkably different properties from the industrial products of the first machine age, a topic which Brynjolfsson and McAfee explore in detail in their book. They define "digital technologies" as "those that have computer hardware, software, and networks at their core" [2, p. 9]. It is within this wider context of digital technologies and the second machine age that this thesis is best understood.

Digital technology is a broad category; therefore, it is useful to narrow the focus to a few key technologies that are driving the development of the second machine age. There are four specific technologies that have particular relevance to this thesis: (1) mobile telecommunication devices, (2) the Internet, (3) positioning technologies, and (4) a wide range of inexpensive yet highly capable sensors, namely microelectromechanical systems (MEMS). We note that these four technologies have converged over the course of a few decades, so that the changes are clearly evident within one human generation (i.e. 20-30 years). All of these technologies came to

[1] By "technological convergence", we mean that a set of technologies has undergone rapid advances simultaneously and thus have become available for technological uptake in combinatorial ways.

a technological crossroads in the late 20th century and early 21st century, so that a child born and raised in the 21st century will have vastly different technological possibilities, compared to one born and raised in the 20th century.

The applications arising from this technological convergence span many different areas, and we have no intention to cover these applications exhaustively in this thesis. Instead we focus on one application area—navigation. The research scope of this thesis will be defined more precisely in Section 1.2.

The first major manifestation of this technological convergence, especially with respect to consumer markets, is the so-called "smartphone," which incorporates or supports all four of the above-mentioned technologies. Looking at the history of mobile devices, it is difficult to say which mobile phone can be considered the first smartphone. The first commercially-available phone with a Global Positioning System (GPS) receiver, the *Benefon Esc!*, was released in 1999. In terms of marketing, the Ericsson R380, released in 2000, was the first mobile phone to be called a smartphone. In terms of the four technologies listed above, the Samsung SCH-S310, introduced in 2005, was probably the first to exhibit all four. The first iPhone was released in 2007, and the first Android phone was released in 2008.

About 64 million smartphones were sold globally in 2006 [3], and by 2008 this number exceeded 139 million [4]. By 2012, there were already more than one billion smartphones in use worldwide [5]. This number is forecast to reach nearly 2.5 billion in 2015 [6]. These devices allow their users to stay "connected" virtually everywhere they go, and consequently anyone can connect to these billion plus users from any networked device, including desktop computers and "land-line" phones—no matter where the user is located or traveling to. Ironically, in many technologically advanced societies, it is now considered a societal and/or behavioral challenge for one to go "off the grid" or "disconnected" for any extended period of time.

It is our view that the smartphone is only the first manifestation of this technological revolution. Many other so-called "smart" devices are soon to follow: "smartwatches" and the use of various wearable sensors may soon become a mainstay consumer habit. In addition, the same technologies that have made smartphones possible and popular are quickly making their way into existing everyday devices, including cars, home appliances, and even toothbrushes. Furthermore, it is not just consumer markets that are being transformed but also many industrial markets, ranging from manufacturing to commercial shipping. It would be naïve to speculate exactly how this revolution will play out in the coming decades, but it is clear that the developments are already changing the lifestyles, habits, and possibilities of people living in the early 21st century, especially those who can afford these (currently) “high-end” consumer devices.

Aside from being a convergence of new digital technologies, is there any unifying concept or principle that is underlying this technological revolution? Some would argue that it is the increased levels of *mobility* that these technologies provide. Others have rallied under the banner of *ubiquitous computing* or *pervasive computing*, which describes the fact that computing devices can now be found nearly everywhere one looks. Certainly these are two important characteristics giving wind to this revolution, but we argue in this thesis towards another underlying principle that provides a common thread and deep insight into how our relationship to these computing devices is changing.

One common development, of course, is the increasing ability of computing devices to fulfill various user desires, e.g. to download large amounts of data at high speeds, to capture or render various high-quality multimedia content, to store and edit content in various ways, etc. What is not advancing or expanding—at least, not at any considerable rate—is the patience or attention span of the users themselves. Therefore, users are expecting (consciously or not) that their devices will “do more” with essentially the same total quantity and quality of human input. Fortunately, however, these devices are rapidly advancing in their ability to know what their users want or need without the user having to explicitly formulate and express these desires to the computer.

It is our view that we are not even close to unleashing the full potential of computing devices to understand their users. In many ways, smartphones and other so-called *smart devices* are not yet “smart”. They have the “brawn” and not the brains, in the sense that they are powerful and capable but deficient in understanding the user’s needs.

The field of study related to how humans communicate with computers and vice versa is known as Human-Computer Interaction (HCI), and this thesis finds relevance in HCI. Essentially, we aim to reduce the need for explicit human-to-computer communication by increasing the ability of computers to understand humans. This is the goal under which this thesis is motivated and focused—to improve our understanding of how computing devices can better understand us and our needs.

The primary method by which this thesis aims to achieve this goal is through *machine learning*. According to Tom Mitchell and co-authors, "machine learning research seeks to develop computer systems that automatically improve their performance through experience" [1]. This is our favorite definition of machine learning among the many found in the literature, but we note that achieving such a system is incredibly difficult. Most methods that go by the name of "machine learning" fail to meet this definition in terms of *automatically* improving performance. Nonetheless, the discipline of machine learning has grown in recent decades, and the set of techniques going by the name of machine learning is indeed very powerful. In many ways, machine learning has become the preferred framework for building up systems that understand users' needs. Some observers may note that such systems exhibit—or at least attempt to exhibit—*artificial intelligence*.

Artificial Intelligence (AI) has been an elusive goal of computer science researchers ever since the term was coined in 1955[2]. Although computers have not yet replicated human intelligence in a general sense, there are many tasks of increasing complexity that computers can already perform equally well or even better than the most gifted, well-trained humans. As detailed in [2], computers have been programmed to beat even the best human players of the game-show *Jeopardy!*, to write corporate earnings previews for *Forbes.com* that are indistinguishable from ones written by humans, and to diagnose breast cancer from images of tissue as good as or even better than pathologists can[3]. Such examples demonstrate the increasing practicality of artificial intelligence, but what about understanding users' needs? Is it possible for a computer or computing system (including various sensors) to know what its user needs or wants before he or she makes any keystroke or swipes any touchscreen? Such a system would be considered by many to exhibit a high level of artificial intelligence.

[2] Although McCarthy is usually credited with coining the term artificial intelligence, we note that its first usage in the literature was a paper co-authored by McCarthy, Minsky, Rochester, and Shannon [7]. Therefore, it is not entirely clear who first came up with this term, and in an interview even McCarthy himself could not recall [8].

[3] To be precise, what Brynjolfsson and McAfee describe is a system, known as C-Path, that helped to diagnose breast cancer and also identified new features of breast cancer tissue that were shown to be good features for predicting survival.

1.2 Research Questions and Scope

The goal stated above is ambitious and open-ended. To narrow it slightly, this thesis aims to improve the state-of-the-art in a computer's ability to understand situations or contexts that humans find themselves in. Mobile computing researchers have adopted the term *context awareness* to refer to this ability. In other domains, such as aviation, maritime, and military domains, the term used is *situational awareness* (or *situation awareness*)[4]. The science related to context awareness (or situation awareness) is vast. Therefore, we have focused on one particular application area, navigation, where context awareness may be applied.

In particular, this thesis will be organized around a central research question:

- What are the benefits and constraints of introducing context awareness in navigation?

Highly-respected navigation researcher Dr. Paul Groves has called context one of the key challenges for the next generation of navigation technologies [9], and we share this view.

A secondary research question addressed in this thesis is:

- How can machine learning be used to build context or situation awareness, in order to solve problems in navigation?

Given that machine learning has been successful in many other related application areas, our hypothesis is that machine learning will prove to be an effective tool for building context awareness for navigation applications.

By focusing on navigation, we have limited the scope of research to a reasonably-sized domain. That being said, improvements in the state-of-the-art in context awareness have wide-ranging applications, and it is our hope that the few applications described in this thesis are seen only as examples and not as end goals in themselves. There are a wide range of rich mobile applications that can be enabled with the aid of context awareness, and researchers have identified context awareness as one of the key open issues in mobile computing research [10].

[4] For consistency, in this thesis we primarily use the term context awareness, but it can be considered synonymous with the term situation(al) awareness.

1.3 Key Issues in Navigation Research

Before we attempt to answer the above research questions, it would be prudent to define what we mean by *navigation* and discuss some of the key issues in navigation research. General principles in navigation will be discussed in more detail in Chapter 2.

No universally agreed definition of navigation exists [11]. In general, the definition of navigation varies by industry. For example, the maritime, aerospace, and road transport industries each have their own views of what is meant by navigation. We mostly follow the delineation of navigation described in [11], which divides navigation into two distinct concepts:

1. the determination of the position and velocity of a moving body with respect to a known reference point.
2. the planning and maintenance of a course from one location to another, avoiding obstacles and collisions.

The first concept is relatively straight-forward compared to the second. Over the years, many techniques have been developed for determining the position and velocity of moving bodies. Today, however, the "gold standard" method for determining position and velocity is the use of Global Navigation Satellite System (GNSS). GNSS receivers, which have in recent years become relatively inexpensive and small, are capable of achieving position accuracy of below 10 m under most circumstances worldwide. Velocity measurement accuracy, depending on the technique used, can be on the order of a few cm/s [12]. The drawback of GNSS is that it requires the acquisition and tracking of weak radio signals broadcast from space. If the view to the sky (from the receiver's perspective) is significantly obstructed by, e.g. structures or vegetation, then the accuracy will degrade, and in the worst circumstances no position or velocity solution will be obtained. For example, in most indoor environments, unless the receiver is near a window or other glass structure, it will be unable to perform positioning or velocity determination.

Therefore, one of the key research issues in navigation research with respect to the first item above is developing methods for position and velocity determination in indoor and highly urban environments. There is not a single positioning technique that works well in *all* environments, and thus solving the problem of ubiquitous positioning requires a hybrid approach. Due to this fact, a ubiquitous positioning system must also be capable of knowing when to utilize GNSS vs. some other method. This is one of the areas where context awareness can benefit navigation.

The latter concept of “planning and maintenance of a course” includes topics such as route optimization, navigation visualization (e.g. turn-by-turn navigation), and hazard detection and avoidance. In particular route optimization is an important topic in navigation, and there are a number of active research topics in this area, e.g. dynamic optimization of road journeys to avoid traffic, safety optimization, fleet management, etc. In certain contexts, “planning and maintenance of a course” may include determining the required maneuvers and associated vehicle parameters for achieving those maneuvers (e.g. in maritime navigation, the ship’s rudder positions and engine power, etc.). Lastly, in the context of modern Location-Based Services (LBS), this latter concept has come to include many ancillary functions that were not part of traditional navigation systems. One example is the search and retrieval of information concerning possible destinations. Especially in the case of navigation being integrated into mobile devices, the dividing lines between “navigation” and various other LBS functions is becoming increasingly blurred. For example, location “check-in” services are being integrated into navigation applications, although a check-in does not strictly fit into the above definition.

Finally, one active research area which combines the above two concepts is Simultanous Localization and Mapping (SLAM), where the goal is to determine a vehicle or pedestrian’s position (and velocity) while at the same time produce a map of an unknown environment.

Given the wide gamut of functions that navigation systems must perform, depending on the application, one of the other key issues in navigation is how can a navigation system gather and maintain a “picture of the world” such that it has complete and up-to-date information about how to best perform all of these functions. In this sense, context awareness is ideally suited for navigation because it can provide this picture of the world.

1.4 Research Methodology

The research methodology adopted for this thesis varied according to several distinct phases. In the first phase, a wide literature review was conducted on the subject of context awareness. The results of this literature review are presented primarily in Section 4.2 but also in the included publications. In the second phase of our research, we aimed to synthesize the results of our literature review and formulate a general theoretical framework for contextual information and contextual reasoning. This work is reflected in Chapter 4 and publications [P1] and [P2]. Finally, the third phase of our research focused on tackling three different research tasks related to introducing context awareness into navigation applications. Determining how to best accomplish these tasks required different research methods, depending on the task, although the first two tasks were more closely linked compared to the third. The tasks were: (1) to recognize the activity of a smartphone user in an indoor office environment, (2) to recognize the mode of motion that a smartphone user is undergoing outdoors, and (3) to determine the optimal path of a ship traveling through ice-covered waters. These tasks are very different from one another, especially the third task with respect to the first two, demonstrating the breadth of problems encompassed by the topic of context awareness in the field of navigation. They also demonstrate wildly different aspects of "understanding users' needs" for different types of users.

The first task provides possible enhancements for a navigation or position tracking system that must work also indoors. As discussed briefly above, reliable and everywhere-available indoor positioning is one of the biggest unsolved problems in navigation. Context awareness is one way to help solve this problem. For example, if the system detects that a user is sitting and working in a static position (e.g. seated at a desk), then it can apply a positioning filter that assumes little or no changes in user position (and perhaps go into a low-power-consumption mode), but when it detects that the user has stood up, it can change the filter to one that assumes greater possibilities for movement. If the system later detects that the user has done some routine activity, e.g. fetched a fresh cup of coffee, it can apply a post-processing filter to refine the position tracking history, perhaps removing outliers or some other desired refinement.

Similarly, the second task is important for navigation because a navigation system designed to work outdoors can adapt and improve its performance based on the motion mode in which it is used, but it would be easier if the user did not have to manually change the modes of the navigation system when he or she transitions, e.g. from walking to driving. In other words, a context-aware navigation system would automatically know that a pedestrian user needs a pedestrian navigation system and a driving user needs a car navigation system; it would adapt itself automatically according to these different needs.

The third task is a rather classic problem in maritime navigation, but surprisingly this function has been and continues to be performed in a manual way (i.e. the ship captain or navigator manually choosing the route based on ice charts, local observations, and experience). It is also becoming increasingly important to find efficient paths through ice-covered waters due to the opening up of northern sea routes, as well as increased wintertime maritime transport in general (e.g. in the Baltic Sea). In terms of understanding the users' needs, this capability means that if maritime conditions change, such that the captain or navigator needs to alter the ship's route (based on changing ice conditions or other factors), an "ice-aware" navigation system could automatically inform the ship's crew that a new route is recommended and even suggest the optimal route to the crew.

For the first and second tasks, we utilized a method from machine learning called supervised learning, which is a quantitative research method. Supervised learning will be discussed in more detail in Section 3.3, but in brief this method uses labeled "training data" to measure the performance of a learning process. In the context these two tasks, the learning task was to recognize activities and motion modes of smartphone users. Thus, the performance on this learning task was measured by comparing the inferred activity/motion with the actual activity/motion using labeled data.

The third and final task was investigated using feasibility assessment, in which we investigated the viability of developing an ice-aware maritime route optimization system. In this research work, we studied the relevant state-of-the-art and proposed a method for performing the desired route optimization. In this sense, our work can be considered as constructive research. The chosen method is based on graph theory and a breadth-first search algorithm. Finally, we implemented the method in software and validated it by comparing the computed routes with historical routes. More details on the methodology for this task are found in [P5].

1.5 Main Contributions

This thesis explores an important link between machine learning and context awareness and exploits this link to demonstrate possible applications in the field of navigation. Firstly, the author has developed a generic conceptual framework for the multi-step processing of raw sensor data into contextual information, which had been largely lacking in the literature. Also, many studies on context awareness either focus on a narrow area of context (e.g. [13] [14] [15] [16]) or do not provide any clear framework or mechanism of how to encode a situation or context in a systematic and comprehensive way (e.g. [17] [18] [19] [20] [21]). This thesis proposes and describes a simple but powerful framework for describing a context in terms of seven key questions, covered further in Chapter 4 and [P1]. Together, these two conceptual frameworks benefit the research community by making the abstract and ambiguous concepts "context" and "context awareness" more concrete and clearly defined, as well as providing a methodological skeleton on which to build context-aware systems.

In addition, this thesis examines three separate use case scenarios or applications of context awareness related to the field of navigation. These correspond to the three tasks described in Section 1.2 above. The remainder of this section describes the key contributions related to these use case scenarios.

Firstly, the thesis presents a probabilistic Location-Motion-Context (LoMoCo) model, combining location and motion context, used to detect human behavior (i.e. activities) in an indoor office environment. The sensors used to detect the human behavior include only sensors available in commercial-off-the-shelf (COTS) smartphones, as well as access points in a wireless local area network (WLAN) used for the positioning component. To our knowledge, this is the first study focused on detecting office-environment activities that utilizes only smartphone-based sensors and standard WLAN access points. This is significant because earlier studies mostly relied on the installation of custom-designed sensors in the office environment or wearable sensors that are not in common use in offices. For example, [22] relies on sensors installed in an office chair and multiple cameras installed in an office room to infer activity. As smartphones and WLAN access points are already widely present in office environments around the world, the results of this research have more potential for widespread application. Our method can be used anywhere within an office building where WLAN signals are present, provided the user has a smartphone.

A problem related to the above topic is the determination of whether a smartphone user is indoors or outdoors. This is important contextual information because the optimal positioning system differs depending on whether the user is indoors or outdoors. Another important benefit of this contextual information is that it can be used to conserve smartphone battery usage. Outdoor positioning systems, namely those based on Global Navigation Satellite Systems (GNSS), are power intensive and should be turned off automatically when the user is indoors. The method described in this thesis for indoor-outdoor determination is, according to our knowledge, the first smartphone-based probabilistic indoor-outdoor method described in the literature. A similar method was published later as a patent application [23].

Next, this thesis includes a systematic evaluation of a large number of machine learning algorithms applied to the problem of detecting "mobility contexts", including consideration of the computational cost of the resulting classifiers, due to their intended use in resource-limited mobile devices. The number of algorithms investigated and applied to this problem is larger than any other previous study, according to our knowledge. Also, most existing studies dealing with mobility context do not consider or evaluate the computational cost of classifiers, so our study is novel in this aspect.

Furthermore, our study is the first research on mobility context to utilize GNSS, accelerometers, and information from Geographic Information Systems (GIS) for the purposes of detecting mobility context[5]. A similar study utilized GNSS and GIS information but not accelerometers [13]. In particular, GIS is an important source of information for detecting mobility context because it can be used to determine proximity to relevant landmarks, such as train stations and bus stops. Most earlier studies do not consider this important source of semantic information, and our research provides strong evidence, as a result of feature selection, that such information improves the context recognition result.

The main contribution of this study was to measure the relative performance of many different types of classification algorithms applied to this particular machine learning problem. Using default parameter values, the best performance was achieved using

[5] This research was first published as a conference paper in 2013 (see [24]). The publication included in this thesis is an extended version of this earlier study.

the RandomForest algorithm. We also studied the influence of parameter tuning for the RandomForest algorithm. After parameter tuning, we achieved an average recall rate of >97.5% for our test data. We are not aware of any other study achieving this level of performance for a comparable classification problem. One limitation of this study is that exhaustive parameter tuning was not performed for every algorithm type. Therefore, it is possible that other algorithms, after tuning, may achieve similar or even superior performance.

Lastly, for the purposes of developing an “ice-aware” maritime navigation system, we developed a novel method for route optimization. Compared to earlier works in this area, our method is only the second graph-based approach to the problem of route optimization through ice-covered waters. Compared to the earlier graph-based approach, described in [25], our method is more computationally efficient, since it uses the A* algorithm rather than Dijkstra’s algorithm. We note that in [25] only a few tens of nodes were considered in the route optimization examples given, so computational complexity was perhaps not an apparent issue. To find truly optimal routes over large distances, however, it is necessary to consider thousands of nodes or more. [26], published later than [P5], also used the A* algorithm. Compared to [26], we incorporated into our method an operational constraint related to ice breaker assistance, whereas the cost function used in [26] did not consider this issue. Another advantage of our cost function is that the cost, expressed in the unit of time, is easier to interpret. The cost function employed in [26] is a linear combination of four different variables, and its physical meaning is difficult to interpret.

An earlier study tackling ice-aware route optimization expressed the problem as a differential equation and used numerical methods to solve it, such as Powell’s method [27]. Such methods, however, do not guarantee a global optimum. Due to the complex nature of ice fields, local minima can be significantly worse than the global optimum. The benefit of a graph-based approach is that shortest-path algorithms exist that can guarantee an optimal solution. The main novelties in our method are the design of a suitable graph structure that provides a reasonable trade-off between realistic modeling of ship motion and computational complexity, as well as the incorporation of ice breaker assistance into the cost function used in optimization.

1.6 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 provides background on the most important principles in navigation relevant to this thesis. Chapter 3 provides an overview of the principles of machine learning relevant to this thesis. Chapter 4 provides a background on context awareness and summarizes our approach to introduce context awareness to navigation research. Chapter 5 provides an overview of the included publications and summarizes the results. Finally, Chapter 6 offers some conclusions that can be drawn from this thesis work and provides some suggestions for future areas of research and development related to context awareness for navigation applications.

2. PRINCIPLES OF NAVIGATION

This chapter contains a concise overview of the principles of navigation. As stated in Section 1.3, no universally agreed definition of navigation exists [11]. We have adopted the definition of navigation described in [11], which divides navigation into two distinct concepts:

1. the determination of the position and velocity of a moving body with respect to a known reference point.
2. the planning and maintenance of a course from one location to another, avoiding obstacles and collisions.

We have divided this chapter into three sections. Section 2.1 describes the navigation system in general. Section 2.2 describes the first part of navigation, i.e. methods for determining position and velocity. Finally, Section 2.3 describes the functions of navigation falling under the second category of "planning and maintenance of a course..."

2.1 The Navigation System

A *navigation system* is a system used to perform or assist in the functions described above. In modern terms, it is a digital system consisting of one or more computational units, memory units, navigation sensors, and in many cases, a user input and display unit. A navigation system may communicate with other systems in a vehicle through clearly-defined interfaces. In many navigation systems, particular those integrated into mobile devices, the system communicates over a wireless network to obtain assistance data, map data, or other ancillary data. Figure 2.1 shows a block diagram of a typical navigation system.

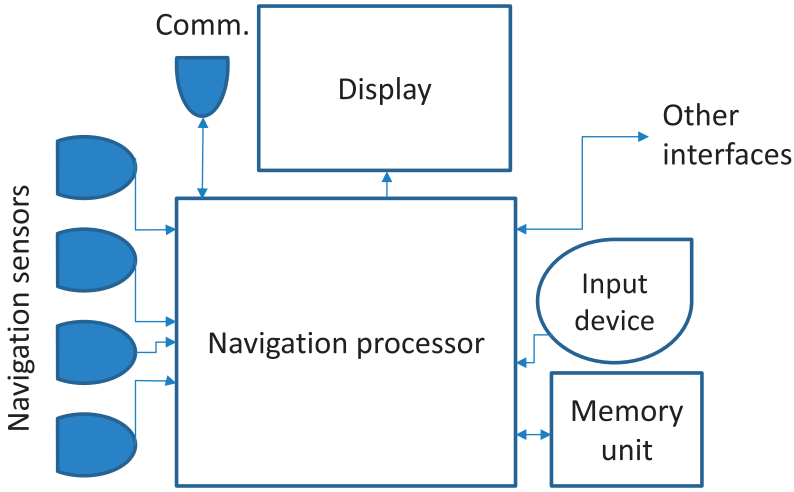


Fig. 2.1: *Block diagram of a navigation system.*

The primary output of the navigation system is known as the *navigation solution*, which consists of (1) the position and (2) velocity of the vehicle or object which is being tracked. A third output, often considered part of the navigation solution, is time. This is due mainly to the fact that GNSS receivers compute time as an intermediate step to determining position. Since a time solution is also needed in many navigation systems, it is natural to include time when speaking of the navigation solution. Finally, some navigation systems also output the vehicle's attitude (e.g. roll, pitch, and yaw), especially in cases where the navigation system includes an Inertial Measurement Unit (IMU).

In some cases, navigation systems may be designed to track objects remotely, but in most cases the navigation system is physically attached to the vehicle or object being tracked. In the case of mobile device positioning, the navigation system is integrated with the mobile device, so there is no separate display, input device, or communications channel. In fact, most navigation systems in mobile devices utilize the host processor and memory of the mobile device, so the integration is indeed very tight. In this context, the mobile device is the object being tracked, but since the mobile device is normally close to the user, we often consider the user to be the object being tracked.

The system requirements for a navigation system vary greatly depending on the application. Different requirements may include accuracy, rate of navigation solution, mass, size, cost, reliability, and various functional requirements (route planning, display, etc.). For example, in aviation and maritime applications, the emphasis is on *integrity*, i.e. ensuring the navigation solution is within the stated error bounds and rapidly informing the user whenever the desired accuracy cannot be ensured. In consumer applications, the emphasis may be on delivering acceptable performance at the lowest possible cost. Due to these varying requirements, there is a great variety of products and solutions in the navigation industry, ranging from navigation systems where almost all the functions are integrated into a single Integrated Circuit (IC) to large, complex, and expensive systems such as those found on submarines.

2.2 Methods for Determining Position and Velocity

In this section we will briefly describe the major methods available for determining position and velocity. Due to space limitations, we cannot cover all such methods, but we will highlight the most important ones.

2.2.1 Trilateration

Today the most commonly used method for determining position is known as *trilateration*. This method, although in use since the late 1940s, was made famous by GPS and other GNSS, which employ this technique. Trilateration uses a set of distance measurements from an unknown position to reference objects to determine the unknown position[1]. The distance measurements are often referred to as *ranges*. In two-dimensional space, a range to a single reference object constrains the unknown position to a disk (or annulus). The width of the disk is determined by the amount of error in the range measurement. Ranges to two reference objects constrain the position to within one region or at most two mirrored contiguous regions (see upper-right of Figure 2.2). By making range measurements to three or more reference objects, the unknown position can be constrained to a small two-dimensional contiguous space, as illustrated in the lower-right of Figure 2.2.

Extending this concept to three-dimensional space, each range measurement constrains an unknown position to within a spherical shell. Two spherical shells

[1] The positions of the reference objects are known to a high level of accuracy.

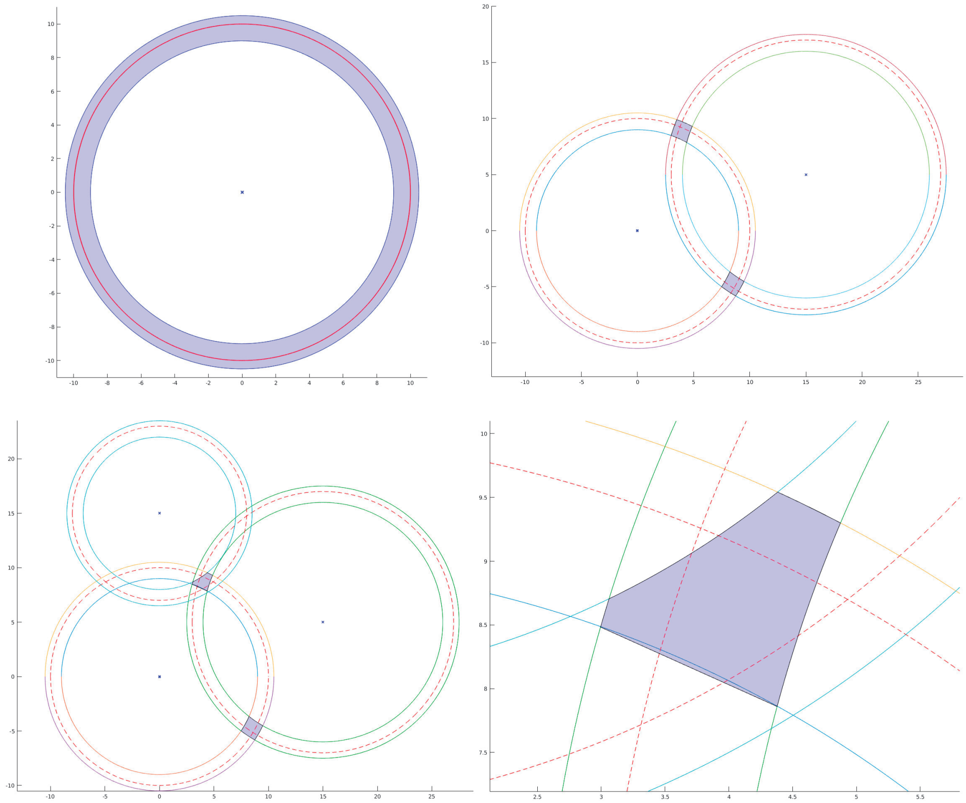


Fig. 2.2: *Example of trilateration using ranging in two dimensions. The final constrained position space is shown in the lower-right image.*

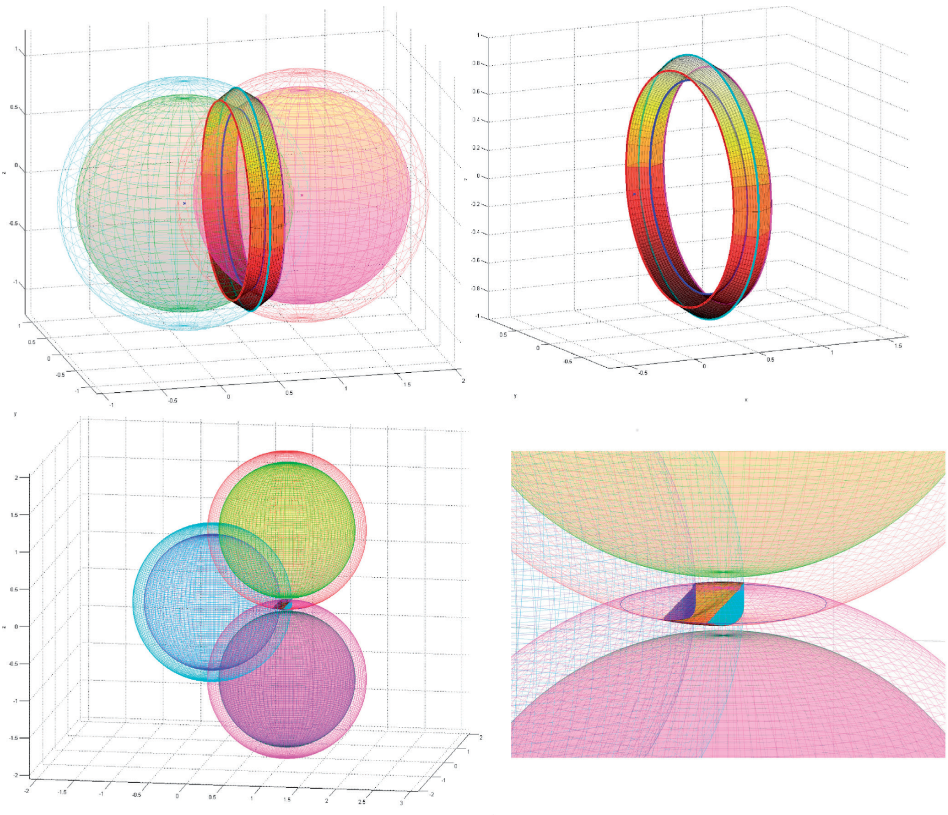


Fig. 2.3: *Example of trilateration using ranging in three dimensions. The final constrained position space is shown in the lower-right image.*

intersect in a three-dimensional shape known as a toroid (see upper part of Figure 2.3, and two toroids, formed from a set of three range measurements, can intersect at most in two mirrored contiguous volumes. In Figure 2.3 for simplicity, we illustrate the case where the two toroids intersect to form one contiguous volume. In practice, if any such "mirror" ambiguities exist, it is usually possible to disregard one of the two volumes because it is, e.g. far from the Earth's surface.

This is the principle upon which GPS and other GNSS are based, although the reality of how these systems are implemented is a bit more complicated. A GNSS receiver measures radio signals from satellites whose position can be determined using orbital parameters. The radio signals contain time signals that can be used to determine the time it took for the signals to propagate from the satellite to the receiver, and because these signals travel at a known, nearly constant velocity (the speed of light), the travel time measurements can be converted to distance. The time measurements, however, contain various biases, the most important of which is the time bias between the GNSS system time and the receiver clock, known as the receiver clock bias. For this reason, GNSS receivers require measurements from four or more satellites to determine a position. The reason that the required number is four and not three, as was depicted above, is because the receiver must also estimate the receiver clock bias.

Because it is well-known that this bias (and other biases) are contained in the measurements taken from the satellites, these measurements are distinguished from the true range and referred to as *pseudoranges*, defined as:

$$\rho \equiv r + ct_u \tag{1}$$

with

$$r = \sqrt{(x_s - x_u)^2 + (y_s - y_u)^2 + (z_s - z_u)^2} \tag{2}$$

where r is the true range, (x_s, y_s, z_s) are the coordinates of the satellite, (x_u, y_u, z_u) are the coordinates of the unknown position, t_u is the receiver clock bias, and c is the speed of light.

Because t_u is common in all pseudorange measurements, its value $\big($along with $(x_u, y_u, z_u)\big)$ can be solved using the following system of equations:

$$\begin{aligned}
\rho_1 &= \sqrt{(x_1 - x_u)^2 + (y_1 - y_u)^2 + (z_1 - z_u)^2} + ct_u \\
\rho_2 &= \sqrt{(x_2 - x_u)^2 + (y_2 - y_u)^2 + (z_2 - z_u)^2} + ct_u \\
\rho_3 &= \sqrt{(x_3 - x_u)^2 + (y_3 - y_u)^2 + (z_3 - z_u)^2} + ct_u \\
\rho_4 &= \sqrt{(x_4 - x_u)^2 + (y_4 - y_u)^2 + (z_4 - z_u)^2} + ct_u
\end{aligned} \tag{3}$$

where $\{\rho_i\}_{i=1}^4$ are the four pseudorange measurements and $\{(x_i, y_i, z_i)\}_{i=1}^4$ are the coordinates of the four reference objects.

Neglected from the above equations are other sources of error which are not necessarily common to all pseudorange measurements, such as ionospheric and tropospheric delays. Although there exist closed-form solutions to the pseudorange equations that take into account these additional error sources, these generally require more than four satellites in order to perform a linear regression and also to estimate the error covariance matrix [28]. For this reason, it is more common to use iterative techniques based on linearization or another estimation algorithm, such as the Kalman filter. More details on how to compute the navigation solution can be found in [29] or [30].

Regarding velocity determination, recall that velocity is the time derivative of position; if the position at two epochs can be estimated and the time between the two epochs is known, then the velocity can be estimated as $d/\Delta T$, where d is the distance between the two positions and ΔT is the time difference. As GNSS receivers also produce a time solution, it is possible to estimate the velocity in this way. This is a simple and occasionally employed approach. Much better velocity accuracy can be obtained, however, by utilizing the *Doppler shift* measurements that a GNSS receiver must inherently measure.

The *Doppler effect* is a change in perceived frequency of a wave due to the relative velocity between the object generating the wave and an observer. Because the satellites are virtually always moving with respect to a GNSS receiver, the receiver must measure this shift in frequency, known as the Doppler shift, in order to maintain track of the signal. The Doppler shift can be converted into a pseudorange rate, and given the pseudorange rates from four or more satellites, the receiver's velocity can be computed. The equations to obtain pseudorange rates can be obtained by differentiating Eq. 3. The details of Doppler-based velocity estimation, however, are beyond the scope of this chapter, and we refer the interested reader to [31]. It will suffice to say that when velocity is estimated using this approach, the accuracy improves several orders of magnitude compared to estimating velocity using simple differencing.

2.2.2 Dead Reckoning

Perhaps the second most common method for position determination is known as *dead reckoning*. Dead reckoning is a form of *relative positioning*, where the current position is estimated iteratively from the known or estimated position of the previous epoch. The basic equations of dead reckoning in two dimensions are the following:

$$\begin{aligned} N_{i+1} &= N_i + s_i \Delta t \cdot \cos\alpha_i \\ E_{i+1} &= E_i + s_i \Delta t \cdot \sin\alpha_i \end{aligned} \tag{4}$$

where (N_i, E_i) are the coordinates in a local geodetic coordinate system at epoch i, s_i is the speed at epoch i, Δt is the time difference between epochs i and $i+1$,

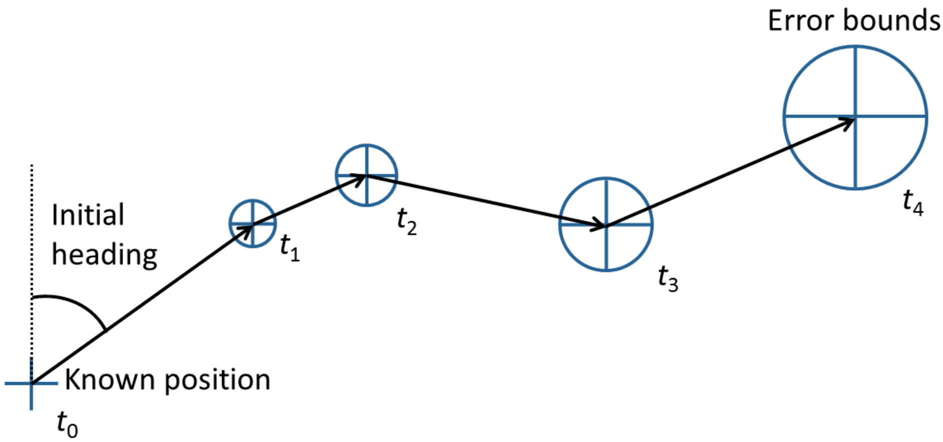


Fig. 2.4: *Illustration of the principle of dead reckoning after [11]. The error bounds of the estimated position increase with time.*

and α_i is the azimuth (or heading) with respect to North. In three-dimensions, the equations are similar, but one must measure also the elevation with respect to the Earth's surface.

One of the main drawbacks of dead reckoning is that the error in the estimated position will increase with respect to time and distance traveled. This principle is illustrated in Figure 2.4.

Nonetheless, dead reckoning is a commonly employed positioning method whenever an absolute positioning method is not available. It is also often used when a positioning solution is needed at a higher rate than that which the absolute positioning system can achieve. For example, it is common that low-end commercial GNSS receivers can only compute a navigation solution at 1 Hz. If higher solutions rates are needed, a dead reckoning method with a Δt less than 1 second may be desired.

There are many measurement techniques available to provide the velocity and heading measurements needed to perform dead reckoning. One of the most commonly used measurement devices is an *Inertial Measurement Unit (IMU)*, which consists of a set of *inertial sensors*, usually consisting of three orthogonal accelerometers and three orthogonal gyroscopes. The use of IMUs for navigation is known as *inertial navigation* and the complete system consisting of an IMU, a navigation processor, and associated hardware and software is known as an Inertial Navigation System (INS). The details of inertial navigation are beyond the scope of this chapter, but we refer the interested reader to [11]. INSs range in performance and cost. The most accurate but expensive INSs may cost in excess of $100,000 and can achieve positioning accuracy of about 1 km after 1 hour of operation[2]. Less expensive INSs based on MEMS technology may cost as little as $100, but they have much worse performance.

Because dead reckoning requires measurements of speed and heading, this method of positioning inherently includes velocity determination.

2.2.3 Positioning Based on Pattern Matching

The final method for positioning that we will discuss in this chapter is positioning based on pattern matching. This method is often known as "fingerprinting" for reasons that will soon become apparent. The basic principle in positioning based on pattern matching is that *a priori* spatially-correlated information about the environment is used to estimate the position based from an observed signal at an unknown location. This is done by matching the observed signal against a database of previously observed signals at known locations or a pre-computed signal-spatial model, thus exploiting the spatial-correlation of the signals. There are many possible methods for performing pattern matching against a database of signals, but a common approach is to find the closest match in "signal space" according to some *distance metric*. For example, consider the following two sets of measured signals:

$$s_1 = \left\{ \begin{array}{c} \text{a=24} \\ \text{b=36} \\ \text{c=14} \\ \text{d=56} \end{array} \right\} \quad s_2 = \left\{ \begin{array}{c} \text{a=13} \\ \text{b=26} \\ \text{c=34} \\ \text{d=55} \end{array} \right\}$$

Each of these sets of measurements represents a signature or "fingerprint" of the location where they were obtained. If the measurements are similar (i.e. distance in signal space is small), then the two fingerprints are considered a match. A commonly-used distance metric is the Euclidean distance, which for this example is given by:

[2] The positioning performance will depend significantly on the type of motion and speeds experienced.

$$d = \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2 + (c_i - c_j)^2 + (d_i - d_j)^2} \qquad (5)$$

where $\{a_i, b_i, c_i, d_i\}$ and $\{a_j, b_j, c_j, d_j\}$ are the four signal measurements in two measurement sets s_i ands_j, respectively. Such signals can be queried against a database, and the best match according to this metric can be retrieved, or a more elaborate algorithm such as k-Nearest Neighbor (kNN)-regression can be used.

In addition to the above approach based on a signal measurement at a single epoch, some techniques utilize a sequence of measurements. For example, [32] uses a time-series of magnetic field measurements to approximate the unknown starting position using a Monte Carlo Localization (MCL) technique. The authors found that in many indoor environments, positioning based on magnetic field could be obtained with sub-meter accuracy after moving about 10 m in a straight line within the environment.

Many possible signals can be used to perform this type of positioning, but some of the more commonly used ones are WLAN received signal strength, other radio signals, magnetic signals, or even images. WLAN-based fingerprinting is perhaps the most common approach in indoor environments.

A drawback to positioning based on pattern matching is velocity determination can only be performed using the simple differencing method described in Section 2.2.2. In cases where the target being tracked is a pedestrian, an alternative approach for velocity determination is *step detection*, where the pedestrian's steps are detected using, e.g. inertial sensors. In this case, the user velocity can be determined within a few 10s of cm/s, provided a good estimate of the user's step-length is available. See [30] for more details on step detection.

2.3 Functions Related to Course Planning and Maintenance

As discussed in Section 1.3, there are a wide range of functions related to "the planning and maintenance of a course from one location to another, avoiding obstacles and collisions." In this section we will describe only a few of the most common and important ones relevant to this thesis.

2.3.1 Route Optimization

Route optimization, also referred to as course planning, path planning, route planning, or simply routing, is the process of determining the optimal route or path that a vehicle or person (or set of vehicles or persons) should take to arrive at a desired destination or otherwise complete some desired task. The definition of optimal can vary greatly depending on the application. For example, one may wish to minimize the travel time, minimize fuel or other costs, maximize safety (i.e. minimize risk), or perhaps optimize other characteristics of the journey such as comfort of the passengers or "providing the most scenic route." Fleet optimization is another topic that is intrinsically linked to route optimization.

Historically route optimization has been primarily a manual task. Navigators or other experienced persons would plan a route based on paper maps and detailed knowledge of the environment and the vehicle. This has been one of the primary roles of "navigators" in maritime shipping and aviation, for example. Algorithmic route optimization, nonetheless, has a rather long history. For example, the "truck dispatching problem" was studied by Dantzig and Ramser more than 50 years ago [33]. Going even further back, the "traveling salesman problem," introduced in its early form in the 1800s, can be seen as a route optimization problem. It is only within the past few decades, however, that computerized route optimization has been realized in practice for navigation applications.

Many different approaches to route optimization have been developed. To tackle route optimization, many techniques from the general field of mathematical optimization can be applied, and the main difference between route optimization methods lies in how the problem is formulated. One approach is to model the environment using graph theory, where different locations are represented as nodes in a graph and the edges or vertices in the graph are associated with a cost to travel between the nodes. This is the approach we took in [P5]. The interpretation of "cost" can vary according to the optimization criteria, e.g. it may defined according to travel time, fuel consumption, risk, or even some multi-modal criteria. Another approach comes from optimal control theory, where the cost is modeled as a function of initial conditions, speed, geometry, vehicle controls, and environmental factors. The problem is then to minimize a continuous-time cost function subject a set of constraints and boundary conditions. Yet another approach is to consider a route as a particular point in a space of $3n$ dimensions, where n is the number of waypoints in the route [27]. Each waypoint has two spatial coordinates and one temporal coordinate[3]. The goal is then to find a point in this $3n$-space that minimizes a cost function. Analytic or numeric methods can be used, but in many cases the cost function is complex and its derivative cannot be solved analytically. The drawback of numeric approaches is that, in most cases, they cannot guarantee global optimality.

2.3.2 Visualization for Navigation

In order to "plan and maintain a course," it may be necessary to provide a visualization of the navigation situation. This is especially the case when there is a human-in-the-loop, but visualization is of relevance even in the case of autonomous robotics, since usually there is a desire for humans to check the computed route or analyze the performance afterwards. Visualizing the navigation situation touches on many topics in HCI, such as cartography, interface design, and geospatial visualization. One topic that is trending in navigation is 3D navigation visualization. Virtual reality and augmented reality are related research areas that have strong ties to navigation.

We mentioned SLAM already in Section 1.3, another trending topic in navigation that is linked to visualization. The goal in SLAM is not only to map and visualize an unknown environment but to use the continuously refined map to constrain the vehicle or person's position. The visualization may consist of a simple 2D map, e.g. the floor plan of a building, but the interesting part from a visualization perspective is how to best represent uncertainty in the map.

Another interesting topic related to navigation visualization and relevant to this thesis is visualization of context or situation awareness. This is a rather new research topic, although some commercial solutions for visualizing situation awareness already exist in the marketplace (e.g. [34]).

[3] For aircraft or spacecraft, this model can be extended to three spatial coordinates, and the problem is therefore solved in $4n$-dimensional space.

2.3.3 Hazard Detection and Avoidance

The last phrase in the latter concept from navigation presented above is "avoiding obstacles and collisions." This is the role of Hazard Detection and Avoidance (HDA). HDA is most prevalent in aviation and spaceflight, but it is also an important system function in maritime transportation and robotics in general. With the increasing research and development in autonomous cars, HDA will continue to grow in importance. Context awareness has clear relevance with respect to HDA, since the existence of hazards can be considered one aspect of context.

Perhaps the most developed application of HDA is in the aviation industry, where such systems are also known as Airborne Collision Avoidance Systems (ACAS). One particular implementation of ACAS standards is the Traffic Alert and Collision Avoidance System (TCAS). Modern aircraft are equipped with several ACAS modules, including Ground Collision Avoidance Technology (GCAT), which warns pilots when they are danger of ground collisions and in some cases may take control of the aircraft to avoid a collision, and also a Midair Collision Avoidance System (MCAS). The International Civil Aviation Organization (ICAO) mandates that larger aircraft are equipped with a TCAS, which consists of a transponder that broadcasts the aircraft's position, speed, and heading. The transponder can also receive and decode radio signals from other aircraft, and these signals are used to warn the pilots if another aircraft comes within a protected volume around their own aircraft. The size of the protected volume depends on the altitude, speed, and heading of the aircraft. If a potential collision is detected, the TCAS of two aircraft can communicate and automatically negotiate the best collision avoidance maneuvers. A related technology, known as Automatic Dependent Surveillance - Broadcast (ADS-B) broadcasts similar information, but it is mainly used for surveillance and tracking of aircraft by the air traffic control system.

Finally, an even more challenging area of HDA research is MCASes specifically designed for Unmanned Aerial Vehicles (UAVs). In the case where there is no human-in-the-loop, an effective design for MCAS for UAVs must be entirely autonomous, and reliable systems are required in order to integrate UAVs into non-segregated airspace. This remains an active area of research and development.

In terms of spacecraft, the main role of HDA is to detect and avoid orbital debris, which may be come from either natural sources, such as micrometeroids or other space material, or human-made sources such as debris from satellites and spent rocket stages. Particularly in Low-Earth Orbit (LEO) orbital debris is increasingly a huge danger to spacecraft. Due to several notable collisions between defunct satellites (or even intentional destruction of satellites using ballistic missiles), LEO is scattered with hundreds of thousands of pieces of debris. For example, the International Space Station (ISS) has had to make more than a dozen evasive maneuvers to avoid orbital debris in its history. Occasionally the crew are required to enter and seal a “lifeboat” when a imminent potential collision is detected, in order to minimize the risk to life. Orbital debris detection is conducted using both ground-based and space-based methods. The US Department of Defense (DoD) tracks more than 500,000 pieces of debris orbiting the Earth using mostly ground-based radar.

Finally, maritime transport is another application where HDA is applicable. In general, large ships are equipped with radar and sonar (i.e. echo sounders) to detect potential hazards, such as other ships, icebergs, ocean debris (such as floating shipping containers), and shallow waters. Since the 1980s, the International Maritime Organization (IMO) began requiring larger commercial ships to carry Automatic Radar Plotting Aids (ARPA), a type of computer system which interface with the ship’s radar to aid in avoiding collisions with other ships. A newer technique more recently required on large commercial ships is the Automatic Identification System (AIS), which works in a similar way to ADS-B. Position, speed, and heading are broadcast to the vessel traffic services (VTS) and to other ships, and the received AIS data are typically integrated into the ship’s Electronic Chart Display and Information System (ECDIS) and radar displays. AIS data also being increasingly integrated into ARPA.

3. PRINCIPLES OF MACHINE LEARNING

This chapter will provide background on the topic of machine learning, whose role in this thesis was outlined in the introduction chapter. We illustrate the main concepts using examples relevant to context awareness.

Our preferred definition of "machine learning research" was also given in the introduction chapter, but it is worth repeating here:

> Machine learning research seeks to develop computer systems that automatically improve their performance through experience [1].

Stated slightly differently, *machine learning* is concerned with developing and analyzing algorithms used by computer systems that automatically improve their performance through experience. An earlier definition, widely attributed to Arthur Samuel, is that machine learning is "the field of study that gives computers the ability to learn without being explicitly programmed"[1]. This definition also implies *automatic* learning, but it suffers from the problem that the meaning of "learn" is not precisely defined.

As is the case in some fields, the discipline known as "machine learning" has drifted somewhat from its original defining aims. This will become more evident later on in this chapter when we describe the major types of machine learning problems that have developed over the past 30+ years.

The chapter is organized as follows. Section 3.1 presents a historical perspective of machine learning. Section 3.2 provides an overview of the modern notion of machine learning. Section 3.3 describes supervised learning, and Section 3.4 describes unsupervised learning. Finally, Section 3.5 provides a few concluding remarks on the topic of machine learning.

---

[1] We have been unable to recover the original source of this quote. Some references cite [35], but the quote is not found in reprints of this article.

3.1 Roots of Machine Learning

Some of the earliest works in machine learning (and artificial intelligence in general) involved computer-based games, such as chess and checkers [36]. Although he did not use the term explicitly, many of the early ideas in machine learning were developed by Claude Shannon in a 1950 paper on computer chess [37][2]. Arthur Samuel further developed these ideas during the 1950s, but his preferred game was checkers [35]. Perhaps the most famous game-playing computer program was initiated at Carnegie Mellon University in 1985 for the game of chess. This project was later transitioned to IBM, culminating in the computer Deep Blue® beating the chess master Garry Kasparov in a six-game match-up in 1997[3]. Nowadays, similarly advanced chess programs can run on a personal computer or even a smartphone or tablet. A full literature review of computer chess is beyond the scope of this thesis, but we refer the interested reader to [38] [39], and [40]. Our intention in this section is use computer chess to illustrate that the early pioneers in machine learning aimed to develop computers that could *automatically* learn to perform a task through experience.

The strategy in developing Deep Blue was to encode into a program both the rules of chess, as well as the tactics and strategies of great chess masters, attempting to cover as many possible situations in chess, also known as *chess positions*, as possible. After a long period of development, Deep Blue eventually succeeded in beating the best human chess players. The main reason for this success was the ability of the computer to evaluate hundreds of millions of chess positions per second, whereas a human chess player relies less on computational power and more on intuition and experience to evaluate the strengths and weaknesses of different chess positions.

Examining the strategy by which Deep Blue was developed, this approach hardly fits the above definition of machine learning. To create Deep Blue required many years of highly “manual” work of programmers refining the set of strategy rules, testing the program against human players, and repeating this process. Thus, it falls short of the aim of *automatically* improving performance. Although this strategy

[2] According to colleagues’ accounts, computer visionary Alan Turing also began considering computer chess during the 1940s.

[3] We note that Kasparov has accused IBM of cheating by letting human players intervene in one of the matches.

was ultimately successful in creating a master chess player, when reading Shannon or Samuel's papers on such subjects, it becomes clear that their aim was not simply solving the direct problem of playing chess or checkers, but rather they used these games as a means of demonstrating a completely revolutionary idea in computer science–that computers might be able to *learn* by themselves.

3.2 Modern Machine Learning

In this section, we describe the modern notion of machine learning, which, as we have already alluded to, has developed into something a bit different from what the early pioneers in machine learning had envisioned. That is, today there is a well-established community of machine learning researchers and practitioners whose focus is not entirely the same as what Shannon, Samuel, Mitchell, or other machine learning pioneers had in mind. Our intention in pointing this out is not to denigrate the discipline of machine learning as a whole but rather to emphasize those aspects of the discipline which fall short of the original goals of machine learning.

Let us first present a few other definitions of machine learning found in recent textbooks on the subject:

> Machine learning is programming computers to optimize a performance criterion using example data or past experience [41].

This definition appears quite close to that of [1], if we assume that "example data" can be generated automatically. This may be true in some cases, but in most methods described in [41], the example data are data that have been manually labeled with the "correct" value relative to the performance criterion that is to be optimized. Although programs using such methods can improve their performance by obtaining more example data, if the example data cannot be generated automatically, then the method would fall short of the definition of [1].

Another recent definition is given by [42], which defines machine learning as:

> a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty (such as planning how to collect more data!).

This definition includes the "automatic" aspect, similar to [1], although we prefer the definition of [1] due to its simplicity. Also, note that there is no reference to improving performance with experience.

We note that some books on machine learning (e.g. [43]) omit to precisely define the concept, perhaps because it has come to encompass many diverse methods. One book goes so far as to explicitly refuse to define machine learning in any principled way:

> The kind of learning techniques explained in this book...are called machine learning without really presupposing any particular philosophical stance about what learning actually is [44].

Mitchell, on the other hand, also provides a precise definition of the concept of learning in the context of machine learning:

> A computer program is said to *learn* from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E* [45].

Continuing with formalisms, many learning tasks can be expressed in terms of learning a mathematical function between the inputs to the task and the desired outputs. In other words, the learning task is to find some optimal mapping between the inputs and the possible outputs. This can be expressed as follows:

$$f : \mathbf{x} \to y \tag{6}$$

where f is a function, $\mathbf{x}$ is a vector of inputs of arbitrary dimension, and y is an output with $y \in Y = \{y_1, y_2, ...y_m\}$, corresponding to the set of all possible outputs (which may or may not be finite).

Machine learning techniques differ mainly in how they express and learn this unknown function $f(\mathbf{x})$, also known as a *model*. They also differ in the form in which y (and therefore the set Y) are expressed. For example, Y may be a continuous range or a finite, discrete set. When the learning task involves a continuous-valued output value, it is called *regression*.

When the output variable is discrete, we call it *classification*, since the possible values generally represent different *classes* or categories. In this thesis, our goal was to output *context*, and most aspects of context (e.g. motion modes and activities) were represented as discrete sets of classes, but in the case of [P5], the ice environment was described mostly with continuous variables.

In some cases, the output of the model may be best represented in probabilistic form. In such cases, the machine learning algorithm actually estimates the conditional probability $p(y|\mathbf{x})$. This distribution, $p(y|\mathbf{x})$, may be intrinsically important to the application at hand, or it may be an intermediate step towards determining the most likely value of y according to:

$$y = \arg\max_{y \in Y} p(y|\mathbf{x}) \tag{7}$$

This is known as the maximum a posteriori (MAP) estimate of y. One of the benefits of estimating $p(y|\mathbf{x})$ is that it provides a measure of the confidence of the output y.

Algorithms designed to learn $p(y|\mathbf{x})$ are known as *discriminative* approaches. An alternative approach is to first learn a model of the joint probability $p(y, \mathbf{x})$ and then condition on $\mathbf{x}$ to derive $p(y|\mathbf{x})$. These are known as *generative* approaches.

Apart from the distinctions regression vs. classification and discriminative vs. generative, there are two main categories of machine learning techniques, based on how the unknown function f is learned or approximated. The first category is known as *supervised learning*. In supervised learning, a "trainer" supervises the learning process. The goal is essentially then to transfer the knowledge of the trainer or supervisor in the form of a mathematical or computerized model. More details on supervised learning will be covered in Section 3.3.

The other main category is known as *unsupervised learning*. In unsupervised learning, the learning process is not guided in any significant way. The goal is essentially to uncover patterns that are implicit in the data but unobvious. Unsupervised learning can be considered automatic, but what can be very challenging in unsupervised learning is to define a notion of performance (recall the definition of machine learning given at the start of the chapter). In this thesis, we have focused primarily on supervised learning, but some research on context awareness also uses unsupervised learning, so we present one example from unsupervised learning in Section 3.4 to illustrate its potential role in context awareness.

3.3 Supervised Learning

As stated above, supervised learning uses a "trainer" to supervise the learning process. In most cases, the trainer has encoded his or her knowledge in the form of *labeled data*, also known as *training data*. In terms of the function f expressed above, the training data consist of input-output pairs $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i$ is an input of arbitrary dimension, y_i is a "labeled" output, and N is the number of training samples, such that $\mathcal{D}$ provides *examples* of values of the function $y = f(\mathbf{x})$. In simple terms, the training data provide sample input data that are *labeled* with the correct or desired output.

It is usually the case that the training data does not exhaustively define the unknown function f. If, however, certain assumptions can be made about the function, then the function might be fully specified by a finite set of training data. In the simplest case, where f is linear and $\mathbf{x}$ is one-dimensional, then only two training samples are needed to specify the relationship between $\mathbf{x}$ and y[4]. Most practical examples of machine learning algorithms, however, are more complicated due to (1) higher dimensionality, (2) non-linearity, and (3) error present in the training data.

Let us consider a simple example from the domain of context awareness. Suppose we would like to develop a smartphone application that needs to know whether the user is walking, running, or standing still (i.e. static). We refer to these as *mobility contexts*. The smartphone has a GPS receiver that can record the user's position and speed, and it also has a three-axis accelerometer that can measure acceleration. Instead of using the raw accelerometer signal, we define a feature from the accelerometer data, known as *dynamic acceleration*:

$$a_d = var(\{\sqrt{a_{xi}^2 + a_{yi}^2 + a_{zi}^2}\}_{i=1}^{N}\,) \tag{8}$$

where $var(\cdot)$ is an operator that computes the variance over some time-series of data (e.g. one second of acceleration data); a_{xi}, a_{yi}, a_{zi} are the accelerations in the x, y, and z directions, respectively, for some given time epoch i; and N is the number of samples in the time-series.

A researcher, Mary, has painstakingly collected a dataset for developing this context-aware application and labeled whether she was walking, running, or standing

[4] Recall that two points define a straight line.

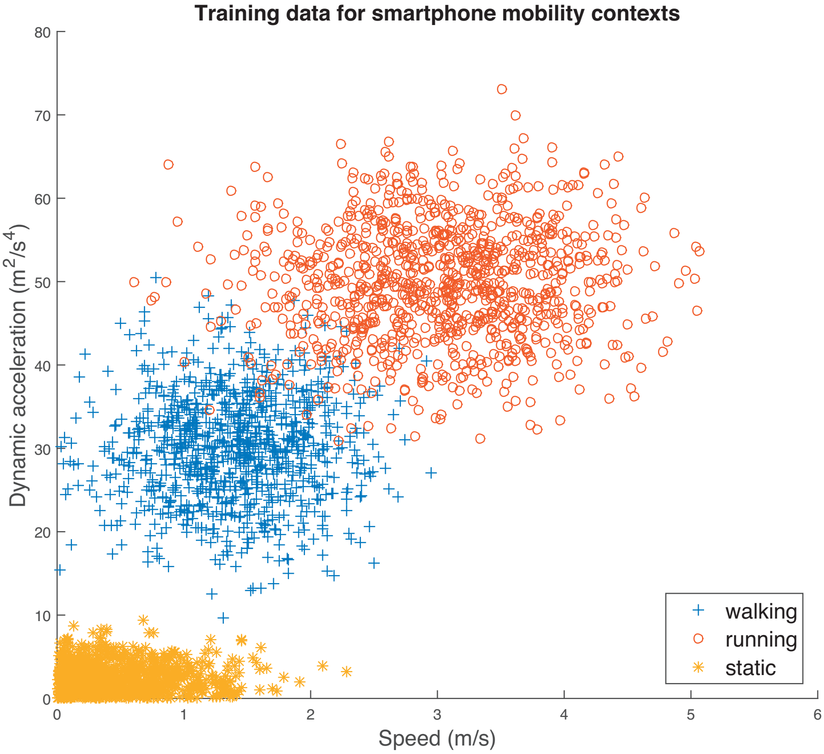


Fig. 3.1: *Example training data for supervised learning. The data are similar to those shown in Table 3.1. Note the partial overlap between the "walking" and "running" classes.*

still. Some sample data are shown in Table 3.1 below, consisting of two dimensions of input data, $(speed^i, a_d^i)$, and the labeled output. In order to keep the size reasonable, only 35 data samples are shown in the table. In Figure 3.1, similar data are plotted, but now we include 1000 samples from each class.

With these data in mind, the goal of supervised learning is to find the function or model f that maps the input data $\mathbf{x}_i = (speed^i, a_d^i)$ to the correct output class $y_i \in \{\text{'}walking\text{'}, \text{'}running\text{'}, \text{'}static\text{'}\}$, such that the number of errors are minimized. In this context, errors are defined as input data that are mapped to the wrong output class, also known as misclassifications.

Table 3.1: *Example data for supervised learning. The data consist of two-dimensional input data from smartphone sensors and a labeled output class.*

ID	Speed (m/s)	Dyn. accel. (m^2/s^4)	Label
1	2.56	21.10	walking
2	0.94	28.78	walking
3	1.24	31.22	walking
4	2.99	36.66	walking
5	1.24	36.43	walking
6	0.64	29.88	walking
7	0.73	34.13	walking
8	1.68	28.56	walking
9	2.72	32.96	walking
10	1.82	38.57	walking
11	2.10	30.70	walking
12	2.80	49.59	running
13	4.01	47.41	running
14	3.10	61.96	running
15	1.98	54.44	running
16	2.33	53.92	running
17	5.48	44.49	running
18	4.14	52.38	running
19	2.69	52.85	running
20	4.73	44.02	running
21	1.22	48.76	running
22	4.88	47.78	running
23	0.40	2.89	static
24	0.92	0.92	static
25	0.36	1.48	static
26	1.16	3.37	static
27	0.00	5.76	static
28	0.28	3.27	static
29	0.60	0.70	static
30	0.45	2.97	static
31	1.44	1.79	static
32	0.11	1.45	static
33	1.36	1.51	static
34	1.06	0.03	static
35	0.81	1.28	static

Based on the above figure, before employing any machine learning, several observations can be made. We clearly see three clusters of data, corresponding to the three mobility contexts. The cluster corresponding to the "static" context is well separated from the other two, but in the case of the "walking" and "running" contexts, there is some overlap. Another important observation is that in the "walking" data, some of the values for speed are very close to 0 m/s. This could be due to errors in the data (i.e. the data from the GPS receiver might have some error) or labeling errors made by Mary. Similarly, the static data contain many points where the speed is non-zero. It is very common with this type of data that some labeling errors are present in the training set. For example, at the transition points between the walking and static contexts, it is difficult to accurately label which data corresponds to "walking" and which corresponds to "static"[5].

Once the labeled data are collected and the desired input features are generated, the next step is *model selection*, in which the optimal model is determined through quantitative performance measures. Correctly measuring the performance requires dividing the labeled data into three distinct sets: (1) the training set, (2) the validation set, and (3) the test set. To aid in model selection, one often performs precursory *data exploration*, in which different features are plotted to study their distribution and how well the different classes are separated. For example, the observations made in the previous paragraph can be considered a type of data exploration (for another example, see [P4]). For example, if it is clear that the classes are linearly separated, then a simple linear classifier [e.g. Linear Discriminant Analysis (LDA)] may perform well.

Three important interrelated concepts should now be introduced: *generalization*, *underfitting*, and *overfitting*. Generalization refers to the idea that supervised learning should "generalize" beyond the specific examples given in the training data. In other words, the goal is not simply to map the inputs to the outputs for the given training data but rather to find a mapping function or model that works well on some yet unseen data. If the goal were simply to fit a function to the training set, then it would be trivial to write a function that performs with zero errors (e.g. a simple lookup table would do the job).

Overfitting refers to the situation where the supervised learning algorithm has

[5] One technique to avoid such labeling errors is to remove these transition points entirely from the training data.

produced a mapping function that follows the training data in too much detail. Keep in mind that every labeled dataset is somehow incomplete and imperfect. If the training results in a function that does not properly take into account the gaps and the noise in the training data, then it will *overfit* the training data and will not generalize well.

It is also possible that a model *underfits* the training data. This usually means that the mapping function is overly simple, for example, using a linear model for data that are inherently non-linear. Therefore, good generalization lies in between the two extremes of underfitting and overfitting.

The goal of learning is more precisely defined as minimizing the *generalization error*, which is the average error rate that will be produced by any future data, and this means finding a model that neither overfits nor underfits. Of course, it is difficult to estimate the true generalization error. This is the reason for dividing the labeled data into three distinct sets. The test set is not used at all in the learning process but is reserved for estimating the generalization error after learning has already taken place[6].

A test set provides a way to measure the generalization error *after* the learning process and to see whether any overfitting or underfitting is occurring, but the question remains: How does one determine the right type of function or model to fit to the training data, i.e. perform model selection? The answer in short is that we use the *validation set* to measure the relative performance of different models and choose the best one for final testing. In detail, the model selection process proceeds as follows:

1. Choose a hypothesis set $\mathcal{H}$ containing different hypothesis function types to be used in model selection. This hypothesis set can be of one particular function class, such as the set of all linear functions or can be of several different classes. The goal is to include within the hypothesis set a class of functions that match well with the underlying data under investigation. This is, however, non-trivial and may require some precursory *data exploration*.

2. Given the hypothesis set $\mathcal{H}$, for each hypothesis class $\mathcal{H}_i \in \mathcal{H}$, use the training set $\mathcal{D}$ to find the best function $h_i \in \mathcal{H}_i$. For example, if $\mathcal{H}_i$ is the set of all linear functions of the form $h(x) = a * x + b$, then this step is equivalent to

[6] The purpose of the validation set will become clear further below.

finding the parameters a and b that best match the training data, according to some linear regression estimator, e.g. the least squares estimator.

3. Now we have a set of fitted functions, each from a different hypothesis class. That is, for each $\mathcal{H}_i$, we have a corresponding fitted function h_i. Let us denote these as $\mathcal{H}_{best-in-class} = \{h_i\}_1^N$, where N is the number of hypothesis classes. The next step is to choose the best h_i from this set. For this, we use the validation set to measure the error rate and choose the function with the lowest error, which we denote h_{best}, and its hypothesis class is denoted by $\mathcal{H}_{best}$.

4. Finally, fit a new function $h_i \in \mathcal{H}_{best}$ using the training set plus the validation set, and measure its error using the test set. Since the test set was not used in the learning process, the resulting error rate can be considered an estimate of the generalization error.

Depending on the amount of labeled data available, and the complexity of the underlying structure in the data, it may be necessary to repeat this process with different divisions of the labeled data into the respective training set, validation set, and test set. The standard technique for this repetition process is known as *cross-validation*. Due to space limitations, we will not cover cross-validation in detail, but it was employed in [P3] and [P4].

So far we have discussed general concepts in supervised learning but not any specific algorithms. Several examples of supervised learning algorithms will be described in [P2].

3.4 Unsupervised Learning

Unsupervised learning is, in many ways, quite similar to supervised learning, except that there are no labeled data. In other words, there are only input data, and the goal is to learn something about the structure or patterns in the input data. In this way, unsupervised learning is very similar to traditional statistical methods, where the goal is to infer a statistical model from a set of data. Many unsupervised learning methods, such as density estimation, come straight from statistics. Others differ only in the name or some other superficial characteristics. Especially in recent years, there are large overlaps between statistics research and unsupervised learning research[7].

Consider again the data presented in Table 3.1 and Figure 3.1. Suppose Mary had not gone to the trouble of labeling the data with the actual mobility context associated with each data sample. We would have then only a two-dimensional dataset of input data, and we could make a similar plot as Figure 3.1, except the legend would be missing and we would also not have the information necessary to label the samples with different colors as in Figure 3.1. The top part of Figure 3.2 shows such a plot.

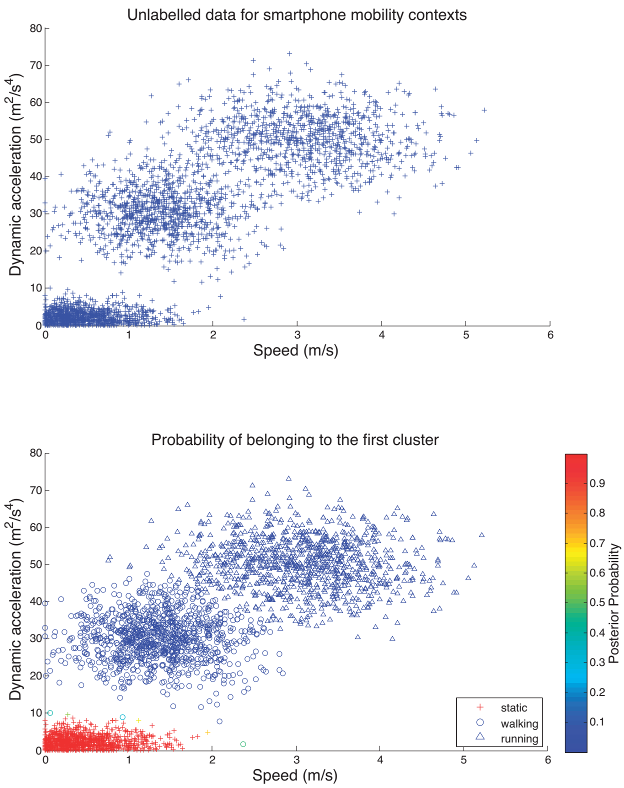


Fig. 3.2: *The top plot shows example input data for unsupervised learning. The bottom plot shows one result from the EM-based clustering. The color of the points shows the posterior probability that the points belong to the first component in a GMM. Data labels are shown strictly for demonstration purposes. In a real situation, no such label would be available to interpret the unsupervised learning result.*

[7] This is also true to a certain extent in supervised learning, but the similarity is more striking in unsupervised learning.

One unsupervised learning task would be to identify different clusters or groups present in the data. Depending on the data and the application, it may or may not be apparent how many clusters are inherently present in the data, so the number of clusters may also be a parameter to determine as part of the unsupervised learning task. There are a plethora of different unsupervised learning algorithms available in the literature that perform clustering. Possibilities include k-means clustering [46], OPTICS [47], and the expectation-maximization (EM) algorithm [48]. In particular, the EM algorithm has its roots in statistics and can fit observed data to an arbitrary statistical model.

To provide an example of clustering, we used the EM algorithm to fit a Guassian mixture model (GMM) to the data that we have previously seen in the top half of Figure 3.2. A GMM is of the form:

$$p(\mathbf{x}|\Theta) = \sum_{k=1}^{K} \pi_k \phi_k(\mathbf{x}; \boldsymbol{\theta}_k) \tag{9}$$

where $\mathbf{x}$ is a random vector, K is the number of components in the mixture model, $\phi_k(\mathbf{x}; \boldsymbol{\theta}_k)$ are normal distributions with parameters $\boldsymbol{\theta}_k = (\boldsymbol{\mu_k}, \mathbf{\Sigma_k})$, π_k are mixing weights satisfying $\pi_1 + ... + \pi_K = 1, \pi_k \geq 0$, and $\Theta = \{\pi_1, ..., \pi_K, \theta_1, ..., \theta_K\}$ is the complete set of model parameters[8].

The EM algorithm itself is a widely-used iterative algorithm used to find the maximum likelihood estimate (MLE) of the model parameters (which we denote with Θ as above) for an underlying distribution $p(\mathbf{x}|\Theta)$ used to model a given dataset, which we denote as $\mathcal{D} = (\mathbf{x}_1, ..., \mathbf{x}_N)$ [49]. The MLE is obtained by maximizing a function Q equal to the expected value of the log-likelihood $\mathcal{L}(\Theta|\mathcal{D}, \mathcal{Y})$, given the observed data $\mathcal{D}$ and the current parameter estimates $\Theta^{(i-1)}$:

$$Q(\Theta, \Theta^{(i-1)}) = E[\log \mathcal{L}(\Theta|\mathcal{D}, \mathcal{Y})|\mathcal{D}, \Theta^{(i-1)})] = E[\log p(\mathcal{D}, \mathcal{Y}|\Theta)|\mathcal{D}, \Theta^{(i-1)})] \tag{10}$$

where $\mathcal{Y} = (y_1, ..., y_N)$ is a vector of latent variables that indicate to which component of the GMM a given data sample $\mathbf{x}_j$ belongs. The latent variables can be expressed in various ways, but perhaps the simplest expression is that $y_j = k$ when $\mathbf{x}_j$ belongs to component k. In the above equation i indexes the current

[8] The notation used for the GMM is similar but not identical to that given in [49].

iteration interval of the algorithm, so $\Theta^{(i-1)}$ represents the parameter estimate from the previous iteration (or the initial estimate, if $i = 1$).

Before applying the EM algorithm to find the parameters Θ of a GMM, one must decide on the number of components K to incorporate into the GMM. As we shall see, each component k in the model will correspond to a cluster in the final clustering result; thus, this step is, in practice, the same as determining the number of clusters, and we can consider K to be a hyperparameter in the estimation problem.

Various methods can be used to determine the best value for K. For low-dimensional data, a practical method is to simply plot the data (as we did in the top half of Figure 3.2) and try to visualize the inherent number of clusters. For high-dimensional data ($D > 3$), this simple approach is not necessarily adequate, nor does it support the goal of automation described earlier. Therefore, a more sophisticated, systematic approach is preferred, such as the one described in [50]. In the interest of space, we assume in this example that the choice of K is already clear, and for these data $K = 3$ seems to be a reasonable choice.

The next step is simply to apply the EM algorithm to determine the parameters Θ of our three-component GMM. A detailed description of the EM algorithm is beyond the scope of this thesis, but here we provide a brief overview.

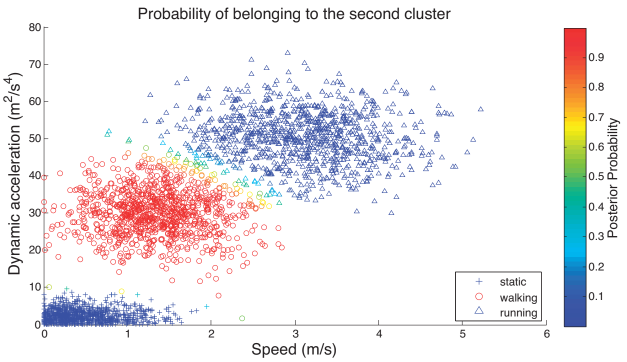
First, EM requires an initial estimate of Θ, and various initialization techniques to provide sensible initial estimates can be found in the literature. A simple approach is to use the given dataset $\mathcal{D}$: e.g. select K random samples to initialize $\boldsymbol{\mu_k}$ and use the covariance matrix of $\mathcal{D}$ for each of the initial K covariance matrices $\boldsymbol{\Sigma_k}$ [51].

After initialization, the algorithm then alternates between computing an expectation function (known as the E-step) and finding the parameters Θ that maximize this function (known as the M-step). At each E-step, the algorithm calculates a new $Q(\Theta, \Theta^{(i-1)})$. In the M-step, an updated estimate $\Theta^{(i)}$ of the parameter set is obtained by maximizing $Q(\Theta, \Theta^{(i-1)})$, according to:

$$\Theta^{(i)} = \arg\max_{\Theta} Q(\Theta, \Theta^{(i-1)}) \tag{11}$$

The algorithm terminates when $Q(\Theta, \Theta^{(i-1)})$, evaluated at $\Theta = \Theta^{(i)}$, converges towards a maximum value (i.e. improvement is below some threshold value ϵ).

Finally, once the parameters Θ are estimated, we can determine the posterior probability that a data sample $\mathbf{x}_j$ belongs to a particular component k of the GMM,



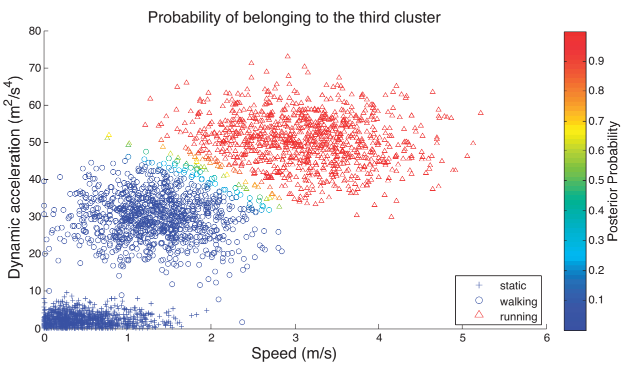


Fig. 3.3: *These two plots continue the results presented in Figure 3.2. The coloring used in the top plot shows the posterior probabilities that the points belong to the second component in a GMM, whereas the bottom plot shows the same for the third component. As in Figure 3.2, the data labels are shown strictly for demonstration purposes.*

according to its so-called "membership weight" [51]:

$$w_j^k = p(y_j = k|\mathbf{x}_j, \Theta) = \frac{p_k(\mathbf{x}_j|\theta_k)\pi_k}{\sum_{m=1}^{K} p_m(\mathbf{x}_j|\theta_m)\pi_m} \tag{12}$$

Recall that each component of the GMM corresponds to a cluster, and therefore the membership weight for a given k is the posterior probability that the data sample belongs to cluster k. The bottom half of Figure 3.2 and Figure 3.3 show the posterior probabilities for our example data, corresponding to membership in each of the three clusters. Note that a dividing line between each cluster can be drawn where the posterior probability reaches 0.5.

Relating this example back to context awareness, we can see from the clustering results that the data can be grouped into three distinct classes, although we have no clear interpretation for these classes in terms of contexts. Nonetheless, we can deduce that two of the classes are somewhat similar relative to the third class (in terms of the two features investigated). Such a result can be useful for context awareness purposes. One purpose is for studying and visualizing the possibility of separating the data into different contexts or situations; if no clusters are evident, it may be very difficult to perform classification with the given features.

Another way to use clustering is as a guide for how to collect training data. A large amount of unlabeled data can be collected and clustered, and then more targeted data collection campaigns can be planned in order to "label" each cluster. Essentially, the clustering results can help one to decide how much labeled data to collect from different segments of the feature space.

Yet another application of clustering is detecting abnormal situations or behaviour. For example, if a particular data sample falls well outside of known clusters of data, then even if the cause of abnormality is unknown, it can be flagged for further investigation. Such techniques are common in applications such as failure detection and security monitoring. Finally, in the context of navigation, clustering and other unsupervised learning techniques can be useful for understanding how different signals might be used for positioning using pattern matching techniques (as discussed in Section 2.2.3).

3.5 Concluding Remarks

Machine learning is clearly a wide topic covering many different concepts and techniques. Our purpose in this chapter was to introduce the most important principles and to elucidate how machine learning can be used to endow computing systems with context awareness. Further examples and details will be given in the included publications. We have emphasized the necessary role of labeled data in supervised learning. We have also demonstrated how a similar result can be achieved through the use of unsupervised learning, although measuring the performance of the result is somewhat problematic.

4. CONTEXT AWARENESS IN NAVIGATION RESEARCH

As stated in the introduction, *context awareness* is the term adopted by mobile computing researchers to describe a computer's ability to understand (i.e. be aware of) the situation or context in which it is operating. In navigation research, of particular emphasis are the *human* context (i.e. the computer *user's* situation) and the *environment* context, but device-specific or vehicle-specific context can also be important to the extent that it can affect the user and his or her goals and ability to achieve them. Examples include: (1) low battery of a mobile device may affect how the user uses the device and even cause him or her to alter plans based on this situation, (2) low fuel level in a car can cause the driver to stop for more fuel, (3) equipment failure in an aircraft can cause the pilot to initiate an emergency landing, etc.

Many definitions of context and context awareness have been proposed, usually reflecting different discipline-specific perspectives. The word context figures prominently in diverse fields including linguistics, psychology, neuroscience, law, and computer science. Due to the great number of definitions, some researchers have used techniques such as latent semantic analysis (LSA) and principal component analysis (PCA) to find the relationships between the many definitions of context [52] [53]. Others (e.g. [54]) have attempted to formalize the concept mathematically.

Let us start by seeing how context is defined in a dictionary. In the Merriam-Webster Dictionary, the word context has two definitions [55]:

1. the parts of a discourse that surround a word or passage and can throw light on its meaning

2. the interrelated conditions in which something exists or occurs : ENVIRONMENT, SETTING

In this thesis, we adopt the second definition. This is because we are not directly concerned with human discourse but rather with conditions of an environment or setting that can be "understood" by computers. Clearly, these two definitions are interrelated—discourse is the way that humans articulate their understanding of an environment or setting. Put in another way, natural language is how humans encode contextual information. In this thesis, we focus on techniques that computers can use to sense, represent, and process context without human intervention. When we refer to context, we refer directly to the conditions in the environment/setting rather than representations of context, such as discourse. As noted in Section 1.2, *situation* can be used as a synonym for context. We see no reason to distinguish between the two terms, although we note that some formalisms make a distinction (see [56]).

With this working definition of context established, we proceed to the remainder of the chapter, which is organized as follows. First, in Section 4.1 we provide two simple frameworks for specifying a context (i.e. the "interrelated conditions") and for contextual reasoning. Next, Section 4.2 provides a review of relevant context awareness literature, paying particular attention to studies relevant to the three tasks described earlier in Section 1.2. Section 4.3 analyzes the differences between our proposed context frameworks and one popular representation of context found in the literature. Finally, Section 4.4 describes the processing chain used to build up context-aware navigation services, beginning with sensing, proceeding up to context recognition and higher-level reasoning capabilities, and finally integrating these capabilities into services.

4.1 Frameworks for Context and Contextual Reasoning

In this section we describe two separate but related frameworks for working with context and contextual reasoning. These frameworks are also covered later in [P1] and [P2], so here we provide only a brief introduction, in order to summarize the ideas and emphasize the author's contributions within the context of the whole thesis.

4.1.1 A Framework for Contextual Information

Because context is such an abstract concept, it is useful to choose some techniques for describing a particular context. These techniques can be used to build a framework for expressing contextual information. The goal of this section is to describe one such technique. We make no claim that this technique or framework is an authoritative one, nor that it is complete in the sense of exhaustively covering the concept of context.

In our view, the goal of context-aware systems is essentially to mimic the way that humans understand and describe situations, contexts, conditions, or events (we use all these terms almost interchangeably, although they may emphasize different aspects, such as fixed versus dynamic elements). According to this goal, we might employ the classic technique of journalism (since journalism is an age-old craft for describing conditions and events), known as the Five Ws: Who, What, Where, When, and Why [57]. This technique can be traced back to the late 2nd century BC when Hermagoras of Temnos defined seven elements of circumstance, which includes (in addition to the Five Ws) “in what manner” and “by what means” [58].

Using these questions as a framework (with a slightly different order), the following provides an example of elements of a particular context:

What: A small gathering of colleagues for lunch

Who: Present are Mary, Philip, George, and Anita

Where: 60.1609°N, 24.5460°E (WGS84); inside the lunch-room of the Finnish Geospatial Research Institute (FGI) in Masala, Finland

When: Tuesday, 10 March 2015 at 11:03AM

Why: Because it is lunchtime, and it is the custom for this group of colleagues to eat lunch together.

In What Manner: Mary’s smartphone is experiencing small, sporadic movements, but it mostly remains in a constant orientation. Mary’s smartwatch is experiencing more dramatic but also sporadic movements. Both sources of motion data are consistent with a user who is sitting and having a casual conversation and/or eating lunch. Multiple human voices are engaged in conversation of an informal and lively manner.

By What Means: All of the above information has been sensed or reasoned by the sensors and software existing in a smartphone and a smartwatch, plus some additional sensor data recorded by a networked node installed in the lunch-room. In this case, the smartphone is a Samsung Galaxy S5 with Android 4.4.2 Operating System (OS), which includes a GPS receiver, WLAN-based positioning engine, Bluetooth connectivity, microphone and audio analyzer, ambient light sensor, accelerometers, gyroscopes, compass, and magnetometers. The smartwatch is an LG G Watch with accelerometers, gyroscopes, Bluetooth connectivity, microphone, and an audio analyzer. It runs Android Wear 5.0.1.

This depiction of the situation is not likely to win a Pulitzer Prize in Journalism, probably because the situation is not particularly interesting. Also, note that it has not been formulated completely into prose but rather is more like a set of notes that a journalist might jot down for later use (except maybe for the latitude and longitude coordinates and the motion description). The "By What Means" section can also be thought of as notes as to the "source" that the journalist might record along with the other information (especially if the account is second-hand).

The seven elements of circumstance can be interpreted as different elements of context, according to the following guidelines:

- "What" usually refers to the *activity context*, that is, what is actually happening. In some contexts, there might be little "action" taking place, but it may also be of interest for some purposes that "nothing is happening".
- "Who" refers to the human characters in the context. When speaking about context-aware mobile devices, the user of the mobile device in question is usually the main character, whereas others in the environment can be thought of as supporting characters. The "who" portion can also be summarized as the *user and social context*.
- "Where" refers to the *location context*. The most important point to note is that location can be expressed in many different ways: geographic coordinates, an address, or some semantic representation such as "the Finnish Geospatial Research Institute" or perhaps more personalized, such as "my workplace". "Where" can also refer to the *environment context*, which includes information about the location where the action takes place. For example, does the action take place indoors or outdoors? Is the location a home, business, etc.?
- "When" is the *time* and *date context*. We need only to be careful about specifying things like time zones. In addition, it may be important to encode some common sense or semantic knowledge about meaningful aspects, such

as "this is after work hours" or "today is a holiday". Time can be specified either as a specific moment, such as in the example above, or as a time or date segment (e.g. 10–20 March 2015).

- "Why" can be thought of as the *motivational context*, e.g. Why is the user doing that? Why is this event taking place? Etc. It could also be appropriate to encode information about whether the context is normal or unusual, as well as an explanation for the unusual events. For example, if a person normally commutes to work along $routeA$, and in the present he or she is driving along $routeB$, then "why" would be a good place to capture the fact that "there was an accident along $routeA$, so the alternate $routeB$ was chosen".

- "In What Manner" is a bit of a "catch all" category. It is used to provide additional details that do not fit nicely into any of the other categories. This is less than ideal for any formal system of context, but rather than attempting to list out all the possible categories of context (which is probably impossible), we believe it is more practical to have an "other" category. One way that we use this category is to capture the *motion context*. This is similar to the activity context, but it is more focused on detailed attributes of the motion. For example, if the current activity of a user is "dancing", then "in what manner" might be used to capture the type of dance and the tempo in which the user is dancing.

- "By What Means", as mentioned above, is to capture the source of the contextual information. It includes information about the devices and sensors used in the context-aware system, as well as the reasoning methods employed.

Further details about this framework are covered in [P1]. As context-aware systems develop, we suspect that our framework may change slightly. It is difficult to anticipate what types of contextual information will become important in the future, but this framework should be broad enough to encompass most types of contextual information. Also, our intention is not to have a rigid framework that constrains all future context-aware systems, but rather it is to provide a rough skeleton upon which to experiment and build more elaborate and detailed context ontologies.

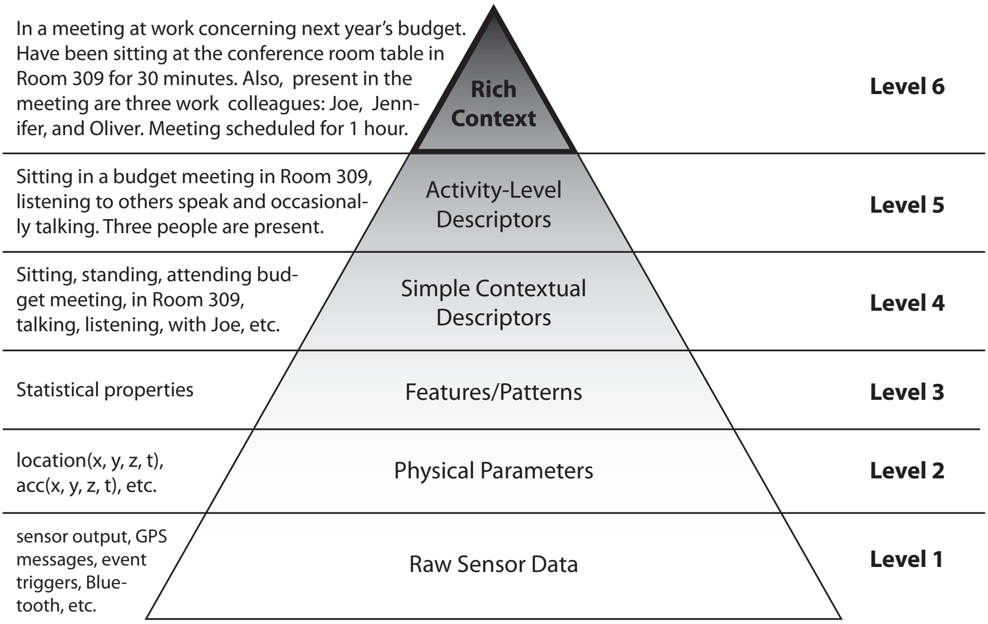


Fig. 4.1: *The "context pyramid" shows the different levels of processing to build context-aware systems, starting from raw sensor data at the bottom and working up to "rich context" at the highest level.*

4.1.2 A Framework for Contextual Reasoning

In this section, we present a framework for *contextual reasoning*, which we define as the process of forming higher level inferences about context from lower-level information. This topic and the associated framework are covered in greater detail in [P2] and partly also in [P3]. We conceptualize the process of contextual reasoning as a pyramid, dubbed the "context pyramid". This context pyramid is shown in Figure 4.1 below. On the left side is an example of contextual information. Inside the pyramid and forming separate layers are different types of data or information related to context. Between each level a step in the contextual reasoning processing chain is inherent. We have not given specific labels to the processing steps, due to the difficulty in generalizing about the processing steps, but we will attempt to describe some common processing steps below.

At the bottom of the pyramid lies the raw input to the context-sensing system, such as sensor data. The difference between the first and second level in the pyramid is that in Level 2, some "pre-processing" of the data may have been performed, such as reference frame transformation or filtering out noise. In the next level of processing, statistical features are extracted from the data, such as mean values or frequency domain features computed from time-series data. The distinction between "pre-processing" and statistical feature extraction can be a bit blurry in some cases, but generally-speaking Level 2 data usually have a clear physical meaning, whereas Level 3 data might have only a mathematical or statistical meaning.

Next, Level 4 is achieved after the Level 3 data are subjected to a function or algorithm that performs contextual classification or in some cases regression. We have proposed that this function or model can be built using machine learning techniques, as described in Chapter 3. This step in the process will be covered further in Section 4.4.2. Level 4 data are in the form of *simple contextual descriptors*, which can be thought of as "atomic" elements of context. They each should belong to one of the seven categories described in Section 4.1.1 above.

Then, Level 5 is achieved by combining multiple simple contextual descriptors into an activity-level description of the context, including the main pertinent contextual details. This process is described in more detail in Section 4.4.3 and again machine learning is the primary tool used to achieve this level in the context pyramid.

Finally, Level 6 combines all available contextual information into a *rich context*. The aim at this level is to approach a description of the context that is indistinguishable from human-written prose. As was the case between Levels 2 and 3, the difference between Levels 5 and 6 can be sometimes blurry. In some context-aware systems there may be fewer processing steps or perhaps more, so even the number of levels should not be taken as dogma. We believe, however, that the general process will always follow the overall trend illustrated in the context pyramid.

As a further example, take the case of an ice-aware maritime route optimization system (i.e. Task 3), which is very different from the example contextual information given in Figure 4.1. At the bottom of the pyramid would lie raw data concerning the ice field, e.g. data from Synthetic Aperture Radar (SAR) or other sources. The first step is to extract physical parameters from the raw data, such as a grid of values for ice thickness and ice concentration, or other system parameters such as the location of an ice breaker. Next, one might perform statistical analysis on the ice data, e.g. applying Gaussian process regression for the purposes of interpolation. Then, one uses features from Level 3 and a ship performance model, in order to estimate the ship's theoretical speed at coordinate $(lat, long)$. Machine learning is one of the possible tools for building the ship performance model.

This type of contextual information is analogous to a Level 4 simple contextual descriptor (i.e. the *in what manner* context). Next, the planned route, computed using the Level 4 contextual information and the route optimization algorithm, is roughly analogous to an Activity-Level Descriptor at Level 5 because it is a higher-level context inferred from lower-level contextual information. Finally, combining the optimal route with other navigational information, such as weather information, maritime traffic information, etc., produces a rich contextual description describing in detail the ship's current situation, i.e. Level 6 at the top of the pyramid.

4.2 Related Studies

There is a vast corpus of context awareness literature. A comprehensive review of recent context awareness literature would require covering hundreds, if not thousands, of different studies. The review found in [59], which only looks at journal articles published between 2000 and 2007, covered well over 200 different studies. Searches of several major databases of scientific publications (e.g. IEEE Xplore Digital Library, ACM Digital Library, SpringerLink) using the keyword "context-aware" each yielded thousands of results. Clearly, these publications will vary greatly in their relevance to this thesis. To help focus on only the most relevant literature, we only include publications related to the three context awareness tasks described in Section 1.4. We also provide a brief overview of the early context awareness research.

According to our review of the literature, the first explicit reference to context awareness was in a 1994 paper by Schilit and Theimer, where they use the term *context-aware computing* to describe software that can "adapt according to its location of use, the collection of nearby people and objects, as well as the changes to those objects over time." [60]. Earlier, however, we can find strong but implicit references to the concept of context awareness. For example, in the 1991 article, titled "The Computer for the 21st Century," Weiser provides a fictional account of a number of different automated or computer-assisted functions made possible by "ubiquitous computing" [61]. Although not specifically highlighted by Weiser, the necessity of these computers to understand context is clearly evident.

Another article, published in 1992 by Want et al., may be the first implementation of a context-aware device described in the literature, even though the authors did not use the term "context-aware" [62]. By the mid-1990s, many different implementations of context-aware devices can be found, including the ParcTab, stick-e notes, CyberGuide, and CyberDesk. By 2001, the research field was active enough to support a special issue of the journal Human-Computer Interaction, which provides an excellent review of the state-of-the-art in context awareness for that time period [63].

Going further back, however, the concept of context has been studied in computer science research for many years. As early as 1963, John McCarthy, one of the "fathers of AI", began developing *situation calculus* as a "formal system in which facts about situations, goals and actions can be expressed" [64]. A situation is defined as "the complete state of affairs at some instant of time", thus, it is roughly equivalent to our definition of context. Beginning in 1987, McCarthy began to consider the concept of context explicitly and attempted to formalize it [65].

Formalisms of context, however, do not appear to have led directly to the realization of any context-aware software or devices, except for perhaps one example, Cyc [66]. The context-aware devices and applications of the 1990s mostly consisted of location-aware devices, and in our opinion, they do not require an elaborate formalism of context. Nonetheless, the work of McCarthy and others pioneers who offered formalisms of context are worthy of mention in the history of context awareness. In particular, we refer the interested reader to [54] [67] [68] [56], and [69]. In addition, [70] and [71] provide excellent reviews of context in artificial intelligence.

* * *

We now review literature specifically related to the three tasks described in Section 1.4.

In terms of the first task, recognizing the activity of a smartphone user in an indoor office environment, there are only a few studies basing their results on smartphone sensors, and they mainly rely on location awareness. For example, one early study described user tests of a context-aware Personal Digital Assistant (PDA) application, which can be considered a proto-smartphone-like mobile device, designed to be used in an office environment [72]. The main source of contextual information was location, but the application also used contextual information from calendar events that the user had previously recorded in the application. For example, if a user scheduled an event at a particular time and marked it as a meeting, then the system would automatically suggest meeting-related services when the meeting time approached and the user had entered the meeting room. The described system, however, was not capable of automatically inferring office activities independently from user-supplied information.

More recently, [73] described a pilot study related to smartphone-based context awareness of “mobile knowledge work”, which may partly take place in office environments. The article, however, mainly discusses the research problem in general, as well as describing the pilot data collection campaign. The article does not go into any detail on the results of analyzing the collected data, nor the planned analysis methods.

If we include studies using non-smartphone sensors to detect office-related activities, we encounter several other relevant studies. For example, [74] used a microphone and a USB camera to detect different office activities and motion patterns, including “rest,” “moving near door,” “conversation,” “nobody around,” etc. They used two machine learning techniques to detect motion and activity: Incremental Hierarchical Discriminant Regression (IHDR) trees and Hidden Markov Models (HMM). [18] similarly used a set of cameras to detect whether an office worker is working alone or in a meeting, or whether the office is empty. For activity detection, they used adaptive background modeling, which makes use of Gaussian mixture densities [75].

Another example is [76], which used a richer set of sensors, including two USB cameras, a pressure sensor, and an acceleration sensor (a Wii remote controller) built into an office chair. This study focused on detecting location and posture context, such as leaning back in the chair, leaning on the desk, or upright sitting posture. They primarily used k-means and kNN clustering to perform context recognition. Similarly, [77] presented a so-called “smart office chair” designed to measure an office worker’s mental and physiological states, such as sleepiness and concentration. The details of the recognition algorithm, however, are not given in the paper.

One study used a combination of mobile phone sensors and room occupancy sensors installed in an office environment [78]. The resulting application, called *WorkSense*, was able to detect when and where meetings and conversations took place. By sensing social interactions, the authors were also able to identify project groups automatically.

Other relevant studies have focused on using wearable sensors to detect different activities, some of which are common in office environments. For example, [79] used accelerometers attached to various parts of a person's body. Although they did not specifically focus on an office environment, some of the activities they addressed are very relevant. These include sitting, standing, walking, writing on a whiteboard, typing on a keyboard, and shaking hands. The authors used a naïve Bayes classifier for activity recognition. Similarly, [80] used wearable sensors to detect several different office-related activities (typing, cleaning a whiteboard, using an elevator, etc.). The authors evaluated the use of kNN and Multilayer Perceptron (MLP) classifiers for activity recognition[1].

Finally, [81] studied context recognition in a meeting room environment, using a suite of sensors including: a microphone array, passive infrared sensors, and an illumination (light) sensor. They detected different states of activity in the meeting room, such as: presentation, monologue, discussion, room idle, people entering room, and room in use. They do not describe their context recognition algorithm in any great detail, but it appears to be a heuristically-defined rule-based algorithm.

The relatively small number of studies related to context awareness in an office environment suggests this is a fairly immature research topic. We can conclude from our literature review that the research in this thesis is rather novel, especially with regards to smartphone-based context awareness for office environments.

* * *

With regards to the second task, recognizing modes of motion that a smartphone user is undergoing outdoors, there are comparatively many relevant studies. Despite this fact, to our knowledge no systematic review of this research area has been published, but a few related reviews are available. For example, [82] reviews literature on the use of GPS to study health-related physical activity. The focus of this review, however, is mostly on assessing health behavior, rather than on methods to detect different mobility contexts. [83] reviews smartphone-based "opportunistic user

[1] MLP is a type of Artificial Neural Network (ANN).

context recognition", which includes some literature on mobility context recognition, and similarly [84] reviews "anticipatory mobile computing". While useful in our review of the literature, none of these reviews focus specifically on mobility context awareness, and they exhibit significant gaps in this regard.

Due to the large number of relevant studies, we will not describe them individually, but pertinent facts from these studies can be found in Table 2.1. We make no claim that this compilation of publications on the subject is exhaustive, but it is, according to our knowledge, the most comprehensive compared to existing literature. We decided to include also studies that utilized wearable sensor modules, since this research is closely linked to smartphone-based research. All of the listed studies used supervised machine learning techniques, except for [85], which evaluated several different unsupervised learning techniques.

Note also that many of the studies investigate not solely mobility contexts but also other contexts, such as Activities of Daily Life (ADL) or various posture contexts (sitting, lying down, etc.). In Table 2.1, we have only identified the relevant motion-related modes. Some other motion-related contexts, such as using an elevator or walking upstairs/downstairs, were investigated by a few studies, but we listed only those modes common to a significant number of studies. For those studies incorporating additional contexts, we have identified these contexts collectively using the symbol "+". Lastly, we point out that some of the included studies investigated indoor motion modes, even though our research task was to study outdoor mobility contexts.

Table 4.1: *Publications related to mobility context. The definitions for the abbreviations used for algorithms/techniques are given in the Abbreviations section provided above. The abbreviations used for the motion modes are as follows: S = static (including standing, sitting, etc.), W = walking, R = running/jogging, B = riding a bicycle, D = driving a motor vehicle, MT = any kind of motorized transport (including car, bus, train, etc.), RB = riding a bus, RT = riding a train/tram/light-rail, and RS = riding a subway/metro. "+" indicates that other motion modes are also covered by the publication.*

Year	Author(s) & Citation	Device(s) Used	Algorithm(s) / Technique(s) Evaluated	Motion Modes Studied
2000	Foerster, Smeja & Fahrenberg [86]	accelerometers	rule-based	S, W, B, +
2002	Lee & Mase [87]	sensor module	fuzzy-rule-based	S, W, +
2005	Lester et al. [88]	sensor module	AdaBoost, NB	S, W, R, B, +
2005	Ravi et al. [15]	sensor module	DT, kNN, SVM, NB, meta-classifiers	S, W, R
2006	Pärkkä et al. [89]	various sensors	rule-based, DT, ANN	S, W, R, B, RB, +
2006	Pirttikangas, Fujinami, & Nakajima [80]	sensor modules	ANN, kNN	S, W, R, B, +
2007	Suutala, Pirttikangas & Röning [90]	sensor modules	SVM, HMM, SVM-HMM, DTS	S, W, R, B, +
2008	Kunze & Lukowicz [91]	sensor modules	DT, kNN, BN	S, W, R, B, +
2008	Jin et al. [92]	sensor modules	fuzzy-rule-based	S, W, R, +
2009	Yang [93]	mobile phone	DT, NB, kNN, SVM, HMM	S, W, R, B, D, +
2010	Reddy et al. [94]	mobile phone	DT, kMC, NB, ANN, SVM, CHMM, DT+DHMM	S, W, R, B, MT,
2010	Pei et al. [95]	mobile phone	rule-based	S, W, +

Table 4.1 – continued from previous page

Year	Author(s) & Citation	Device(s) Used	Algorithm(s) / Technique(s) Evaluated	Motion Modes Studied
2010	Frank et al. [96]	sensor module	NB, BN, HMM	S, W, R, +
2011	Stenneth et al. [13]	mobile phone	DT, NB, BN, RF, ANN	S, W, B, D, RB, RT
2011	Pei et al. [97]	mobile phone	DT, BN, SVM	S, W, +
2011	Susi, Borio, & Lachapelle [98]	sensor module	DT, NB, kNN	S, W, R, +
2012	Bancroft et al. [99]	sensor modules	NB, rule-based	S, W, R, B, MT, +
2012	Anguita et al. [100]	mobile phone	SVM, HF-SVM	S, W, +
2013	Guinness [24]	mobile phone	20 different algorithms	S, W, R, D, RB, RT, +
2013	Susi, Renaudin, & Lachapelle [101]	sensor modules	DT	S, W, +
2013	Feng & Timmermans [102]	sensor module	BN	W, R, B, D, RB, RT
2013	Hemminki, Nurmi & Tarkoma [103]	mobile phone	HMM, Adaboost, meta classifier	S, W, RB, RT, RS
2014	Stenneth [104]	mobile phone	NB, BN, DT, RF, ANN, meta-classifiers	S, W, B, D, RB, RT
2014	Xia et al. [105]	mobile phone	SVM	S, W, B, D
2014	Elhoushi et al. [106]	mobile phone	DT	W, R, B, MT
2014	Parvainen et al. [107]	mobile phone	DT, SVM, MAP	S, W, R, B, MT
2014	Yu et al. [14]	sensor module + mobile phone	DT, AdaBoost, SVM	S, W, R, B, MT
2014	Kwon, Kang, & Bae [85]	mobile phone	GMM, kMC, HIER, DBSCAN	S, W, R, +

Table 4.1 – continued from previous page

Year	Author(s) & Citation	Device(s) Used	Algorithm(s) / Technique(s) Evaluated	Motion Modes Studied
2014	Sankaran et al. [16]	mobile phone	rule-based, GPSAR, FMS	S, W, MT
2014	Chiang, Yang & Tu [108]	mobile phone	DT, kNN, NB, SVM	S, W, R, B, D, +
2015	Yu & Cho [109]	mobile phone	DT, SVM, ANN	S, W, R, D, RB, RT, RS

* * *

Regarding the third task, determining the optimal path of a ship traveling through ice-covered waters, only a few highly-relevant studies are available in the literature. The earliest is [27]. As discussed briefly in Section 1.5, this work expressed the route optimization problem as a differential equation and used numerical methods to solve it, namely Powell's method. As a result, their system cannot guarantee that the computed route corresponds to a global optimum. Similarly, [110] uses a genetic algorithm for route optimization in ice-covered waters. Genetic algorithms have better capabilities to escape from local minima, but they still do not guarantee a global optimum.

The first study to adopt a graph-based approach is [25][2]. The authors present a method for ice-aware route optimization that uses Dijkstra's algorithm to find the optimal route. In the examples given in the paper, only a few tens of nodes were shown for the sea area under consideration. For such small graphs, Dijkstra's algorithm is tractable, but for larger graphs with many edges, Dijkstra's algorithm does not scale well. To overcome this challenge, [26] uses the A* algorithm, which uses a heuristic to guide the search process. Our method, published prior to [26], also uses the A* algorithm. One difference between [P5] and [26] is that the cost

[2] As reported in [110], [111] also used a graph-based approach, but it is only available in Korean. [25] appears to be an extension to [111].

function we developed takes into account possible ice breaker assistance. [26] does not consider this case.

Another related study is [112]. While it does not consider route optimization explicitly, the author investigates ship performance in varying ship conditions. The results are thus applicable to route optimization. Other examples of research in this area include [113], [114], and [115].

Lastly, the project Ice Forecast and Route Optimization (IRO-2), discussed in [116], is very relevant to this topic, but scientific results are not yet available publicly. A preliminary publication reports on tests of the project's prototype ice-aware ship routing system, but the authors only describe performance of the ice forecast model, not the routing system itself [117].

4.3 Analysis of Proposed Frameworks for Navigation Research

In this section, we analyze the proposed frameworks described in Section 4.1 in comparison to an existing definition of context found in the literature and describe why the proposed frameworks are particularly well-suited for navigation research. One of the most similar categorical frameworks of contextual information, compared to ours, can be found in [118]:

> Context-aware applications look at the who's, where's, when's and what's (that is, what the user is doing) of entities and use this information to determine why the situation is occurring.

This framework for context overlaps ours in terms of the first four questions (what, who, where, and when), however, it does not treat *why* as a "first-class citizen" of context but rather as an output of a context-aware application. Our view is that *why*, i.e. the motivational context, is an important aspect of context. It is true, however, that it is often more challenging to determine the motivational context compared to other aspects of context, but *why* can also be an input to higher-level context reasoning processes, so we see no need to treat it differently compared to other aspects of context. Earlier we gave an example of the why context from navigation: that "there was an accident along $routeA$, so the alternate $routeB$ was chosen".

Another example could be taken from an autonomous driving application, where a car owner might in some cases like to drive above the speed limit. Under normal circumstances, perhaps the autonomous driving system would not allow this, but, for example, in an emergency this limitation might be removed. The *why* context would be the natural category in which to encode this kind of information regarding the reasons for operating in a particular mode.

In addition to this difference, the framework of [118] lacks two other categories from our proposed framework: "In What Manner" and "By What Means." Earlier we described "in what manner" as a kind of "catch all" category. The justification for having such a category is that there often arise cases where a piece of contextual information does not fit nicely into any of the other categories, but these examples are scattered such that it would not be desirable to have many distinct categories to cover each of these cases. An example from navigation that comes to mind is:

What: Aircraft in flight between JFK and LAX
Where: 38.6272°N, 90.1978°W, FL310 (i.e. about 31,000 feet altitude)
In What Manner: Flying at 805 km/h and experiencing heavy turbulence

One could argue that particular context categories could be added for specific applications, such as aviation, but our goal in enunciating this framework is to produce a general, flexible framework that can be used consistently in many, if not all, applications.

Finally, the "By What Means" category captures the sources of the contextual information. In this way, it somewhat of a "meta-context" category, since it contains information about the context information and not about the context itself. In a certain sense, however, the sources of contextual information are important aspects of context as well. For example, we propose to include in this category information about the devices and sensors used in the context-aware system, which we argue are an important part of context. This is certainly the case in navigation research, but we believe it would also be the case in other areas of context awareness research. It is not clear in which category this type of information would be included in the definition of [118].

4.4 How to Sense and Use Context for Navigation Research

In Section 4.1 we mainly discussed context and context awareness at a general theoretical level. In this section, we will describe how we approached the problem of sensing and using context in navigation research at a more detailed, practical level. The sub-section structure follows roughly the levels in the previously described "context pyramid", although we have combined some levels in the interest of space. In the last sub-section, we further enforce the motivation for introducing context awareness to navigation research.

4.4.1 Sensing for Context Awareness in Navigation

As stated in the introduction to this chapter, this thesis focuses on techniques that computers can use to sense, represent, and process context without human intervention. Thus, the goal is to "sense" context using automated means. This means utilizing various sensors that can measure the environment and other aspects about the situation in which a device and the user of the device find themselves.

When applying these techniques to navigation, there are a few obvious choices of sensors that can help determine the context. For example, sensors that can determine the position and velocity of the user or vehicle in question are natural choices (e.g. GNSS receivers). Other sensor choices, however, may be less obvious. For example, we stated in the introduction that a navigation system can perform better by recognizing the mode of motion a user or vehicle is undergoing. It may have one mode with a navigation algorithm optimized for when a user is driving and another mode optimized for when the user is walking. The choice of which sensors to use to recognize these different motion modes is non-trivial. In this thesis, we have adopted a practical approach of choosing the sensors that are already available in smartphones (i.e. [P3] and [P4]). In general, however, the choice requires a careful understanding and consideration of the requirements of the navigation application, and the researcher must investigate the unique attributes of each context that must be recognized. A period of experimentation with different possible sensors may be required.

Also, specific sensors often have different options in terms of sampling rate, sensitivity, resolution, or other parameters. Each of these parameters may have an impact on the final context-aware system. In resource-constrained applications, such as in mobile devices, one must not only consider the context recognition performance of the system but also other factors, such as power consumption, memory requirements, etc. In this thesis (namely [P3] and [P4]), we were mainly constrained by the rate at which sensor data could be recorded on the smartphones used in the research. When we attempted to record data from all of the available sensors at the highest possible data rates, the data collection software would often become unresponsive and occasionally also crash. Thus, through experimentation we obtained a balance between data recording rates and stability of the software.

In addition to the choice of sensors and sensor parameters, one must determine what types of features to generate or "extract" from the raw sensor data (recall Level 2 and Level 3 of the context pyramid). In this regard, existing literature dealing with similar context awareness areas is one of the best guides, but of course we aim also to find novel features not found in the state-of-the-art. In this thesis, for example, we use features such as the distance of the user to the nearest train station (computed using the user's current location and the locations of train stations available in a GIS database), capitalizing on the fact that train journeys virtually always start and end at train stations.

Discovering novel features to use in context awareness requires ingenuity on the part of the researcher and also experimentation. It is often not possible to determine the most useful features for context awareness *a priori*. Only after a process of feature selection can the value of different features be evaluated quantitatively (see [P4]). For example, some features that we envisioned would be useful for distinguishing between motion modes turned out to improve the performance negligibly.

4.4.2 Motion, Environment, and Activity Recognition

In navigation applications, the motion ("in what manner"), environment ("where"), and activity ("what") contexts are three particularly important aspects of context awareness. In most cases, these cannot be determined directly from sensors. Instead some type of model must be adopted in order to interpret the sensor data and arrive at the desired contextual result. Continuing the example from the section above, a navigation system that has different modes for driving, walking, etc. must adopt a model for each of these motion modes, where sensor data (or features from the sensor data) are the inputs to the model, and the output is the most probable motion mode. Determining the best model to use for recognizing different contexts is, in our view, one of the most challenging aspects of context awareness research. We have investigated the use of machine learning techniques for building and optimizing models of this type.

Recalling Chapter 3, the process of model selection must be performed to find the best model for inferring context from sensor data. This process was employed in [P3] and especially in [P4]. Different types of models perform differently in different domains. The "no free lunch theorem," discussed in [P2] states that no single machine learning algorithm performs better than any other across all problems. Thus, model selection is unavoidable if one wants to optimize the performance of the model.

We again emphasize that thorough data exploration can help guide the selection of model hypotheses and also to a certain extent in feature extraction and selection. By plotting different pairs of features, one can visualize how the context classes are separated. When the feature dimensionality is high, however, it is difficult to visualize class separation in this way and techniques from unsupervised learning may be more effective. For example, using algorithms like DBSCAN or OPTICS, one can generate two-dimensional plots representing clusters of data of arbitrary dimension.

Another issue that must be considered at this level in the processing chain is how to divide up the context space into distinct classes. We will return to this subject in Section 6.4.3, but for example, in the case of mobility context, there are many possible sets of mobility modes that can be used to divide up this aspect of context (e.g. motorized transport can be grouped together as one context, or different forms of motorized transport can be treated as separate mobility contexts). We argue in Section 6.4.3 that some level of standardization in this regard is sorely needed in context awareness research. The correct choice, however, really depends on the application, i.e. how the context results will be used. Clustering or other unsupervised learning techniques may also help guide the choice because they can reveal how feasible it is to separate different class sets with the available data.

4.4.3 Higher-level Contextual Reasoning

In addition to the type of modeling approach described above, it may be the case that higher-levels of contextual reasoning are needed, where the inputs to the contextual reasoning model consist of outputs from a lower-level model. For example, in [P3] we used the output of motion mode classification, together with location, in order to reason about higher-level activity contexts from a workplace environment, such as "fetching coffee," "having lunch," "working," etc. In some context awareness research, activity recognition is not performed in hierarchical fashion, but our research has shown this to be an effective strategy. In [P3] it not only produced promising classification performance, but this hierarchical approach has the added benefit that the results at the separate layers are more easily interpreted and can be used independently for different purposes.

The methodology used for higher-level contextual reasoning is much the same as presented above, but the major difference is that the labeled data consist of outputs from a lower-level process of context inference together with the desired final output. It will often be the case that some of the input data to this higher-level process contain errors propogating from the lower-level processes, but these errors can be treated in the same way that noise from sensor data is treated. Given enough training data, machine learning algorithms are generally able to minimize the effects of such errors.

4.4.4 Using Context in Navigation Services

It is important to note that context awareness is not an end goal in itself, but rather we would like to use context awareness to improve various navigation services. In order to demonstrate the usefulness of context awareness in a navigation service, [P5] integrates context awareness into a maritime route optimization system. The specific aspects of context that are utilized in [P5] are the environment context, in terms of information about the sea and ice environment, the location context, the speed of the ship, and contextual information about the icebreakers (i.e. icebreaker waypoint locations). The resulting route optimization system is said to be "ice aware," since it especially takes into consideration the current ice conditions and icebreaker information.

The main benefit of introducing context awareness in this particular case is that route planning can be performed in a more automated manner. Currently, maritime routes are mostly planned manually by experienced navigators. Especially in ice conditons, the navigator must integrate many different sources of information in various formats. This is a difficult task for a human to perform and one that is particularly well-suited for computers. The main challenge is incorporating into the route optimization system many different operational considerations that affect the route planning functions, such as safety, economic factors, or maritime traffic conditions. By virtue of the general nature of our context framework, all of these aspects can be considered as part of context in one way or another.

The results described in this thesis are merely a starting point in integrating context awareness into an ice-aware route optimization system, and much further work is needed before the system can be implemented operationally or commercially. Future work in this area is discussed briefly in Section 6.4.1.

The details of how to integrate context into navigation services will vary greatly depending on the requirements of the service, but in navigation there are several aspects of context that are particularly important. A few, such as motion, environment, and activity, have been mentioned earlier, and this thesis emphasizes those aspects. Other obvious examples include location, speed, and conditions of the vehicle (if applicable). In road transportation, traffic conditions are increasingly being incorporated into navigation services. Traffic can be considered as part of the environment context. Another important type of contextual information is semantic information about the location, such as whether a particular area is a commercial, residential, or recreational space. Many databases containing such semantic information have been built in recent years, and these are increasingly available as Application Programming Interfaces (APIs) for use in, e.g. mobile applications.

5. OVERVIEW OF PUBLICATIONS

This chapter provides an introduction and overview of the five publications included in this compendium thesis. Two of the publications, [P1] and [P2], are excerpted from the book *Geospatial Computing in Mobile Devices*, published by Artech House in 2014. Two others, [P3] and [P4], were published in the peer-reviewed open access journal *Sensors*. The final publication [P5] was published in the 2014 Proceedings of the Position Location and Navigation Symposium (PLANS), jointly organized by the Aerospace and Electronics Systems Society (AESS) and the Institute of Navigation (ION). AESS is an affiliate society of the Institute of Electrical and Electronics Engineers (IEEE).

The remainder of this chapter is organized as follows: Section 5.1 briefly summarizes the publications. Section 5.2 maps the included publications to different research areas. Section 5.3 describes the author's contributions to each publication.

5.1 Summary of Publications

Now, we briefly summarize the included publications.

[P1] presents the concept of context awareness at a conceptual level, as well as tracing its historical development. It presents a conceptual framework for describing context, adapted from the seven elements of circumstance, first introduced by Hermagoras of Temnos in the 2nd Century BC. Our main motivation was that we were not satisfied with any of the existing frameworks in the literature in terms of being comprehensive, simple, yet flexible. The publication also aims to show at a practical level how different aspects of context awareness can be implemented in a mobile device. Examples are given for the Android OS.

[P2] presents the concept of contextual reasoning, which is defined as the process of forming higher-level inferences about context from lower-level information. From this definition, we elaborate a conceptual model of contextual reasoning, which we call the "context pyramid". The context pyramid describes contextual reasoning as a series of processing steps at different levels, starting with raw sensor data at the base of the pyramid and working up to the peak of the pyramid, where rich context is realized. We argue that machine learning is an ideal technique available for contextual reasoning, and we provide several examples of different machine learning techniques, such as naïve Bayes' classifiers, Hidden Markov Models (HMM), Bayesian Networks, and Support Vector Machines (SVM).

[P3] combines indoor positioning technologies and smartphone sensors to detect different human activities in an office environment. We provide a real-world implementation of the context pyramid on a smartphone, resulting in a contextual reasoning capability, which we call the "cognitive phone". The key technologies we utilize include ubiquitous positioning, motion recognition, and human behavior modeling. We combine these technologies into a single probabilistic model, which we call the LoMoCo (Location-Motion-Context) model. In this paper, we demonstrate the feasibility of the fifth level in the context pyramid—Activity-Level Descriptors.

The location accuracy we achieved using WLAN-based indoor positioning was about 2–5 meters, depending on the type of space. The positioning performance was better in corridor areas and the worst in lobby areas. We used Received Signal Strength Indicator (RSSI) as the observable and used the pattern matching or fingerprinting approach for positioning. Training data were collected to estimate a Weibull function representing the WLAN signals in the environment. Finally, for position estimation we used the Histogram Maximum Likelihood algorithm.

We also describe our implementation of a probabilistic method for indoor-outdoor detection based on GPS and WLAN signals. For motion recognition, we used the techniques of supervised learning, described in Chapter 3. We performed extensive feature selection, using the sequential forward selection (SFS) algorithm, from thirteen different features derived from the smartphone sensors. Finally, we evaluated several different supervised learning algorithms such as decision trees and LDA, but the best performance was achieved using a Least Squares-Support Vector Machine (LS-SVM) classifier, which produced an average recall rate of 92.9%. The most common errors were confusion between "sharp turning" and "gradient turning," which is not surprising because these motions are very similar.

Next, we used Bayesian inference to determine the most probable activity class, using the location and motion modes as inputs. Locations were represented discretely at the room-level. The detected activities included “fetching a coffee,” “fetching water,” “taking a break,” “having lunch,” “working”, and “undefined context,” which included any activity not otherwise defined. Average recall rate for all contexts was 90.3%. The most common misclassifications were when “undefined contexts” were misclassified as one of the other defined activities. This is not surprising given that the training data collected for “undefined context” consisted of various activities, such as fetching paper from a printer or using a toilet. Our results suggest that it is a challenge to build a strong classifier for a “none of the above” type context class.

Finally, we measured the battery drain of the smartphone that polled sensor data at a high rate, similar to the rates used in the above research. We varied the set of sensors that were turned on to see their relative influence. Less than 5% of the battery was used after 40 minutes with only the inertial sensors turned on. The most significant battery drain was when GPS was turned on, in which case around 20% of the batter was drained during 40 minutes. These results support our view that GPS should only be used when necessary in outdoor environments. From these results, the importance of “indoor-outdoor” context awareness is also supported.

[P4] investigates the use of smartphone sensors, geospatial information, and machine learning to sense mobility contexts, including walking, running, driving and using a bus or train. Our aim was to evaluate techniques that could be used in real-time or near-real-time (<5 s). We also measured the computational complexity of the resulting classifiers because this impacts smartphone battery usage.

It is important to point out that for this application, we are not interested in the time complexity of the training phase but rather the testing phase. This is because training can be done offline where time and power resources are not as constrained. Although it is insightful to consider the time complexity of the testing phase analytically, it is not possible to compare the computation time of different classifiers using pure analysis. This is because the time complexity of different classifiers depends on parameters specific to each classifier, and in some cases complexity is dependent of the training data. For example, in decision trees, the time complexity is linear with the number of non-leaf nodes in the tree. Since the size of the tree is dependent on the training data, it is not possible to analyze the time complexity without first building the tree with training data. For this reason, we measured time complexity computationally for our particular data set.

We investigate a wide range of supervised learning techniques for classification, including decision trees (DT), support vector machines (SVM), naïve Bayes classifiers (NB), Bayesian networks (BN), logistic regression (LR), artificial neural networks (ANN) and several instance-based classifiers (KStar, LWL and IBk). A total of seven features were extracted from two different smartphone sensors (GPS and accelerometers). One of the most novel features used was the distance of the user from the nearest train station or bus stop. This was computed using locations from a GIS database. We performed feature selection to identify the most important features from our dataset for detecting mobility context. These results showed that all features except "speedChange" were useful for detecting mobility contexts. In particular, the GIS features appear to be very useful for detecting mobility context, e.g. performance using one classifier improved from 93.8% to 97.1% when the GIS features were added. Individually, the best performing feature was speed, which is perhaps not surprising in the case of mobility context.

In terms of parameter tuning, we focus on the best performing classifier, RandomForest, which is a type of ensemble decision tree algorithm. Using a hold-out set, we tune its parameters to find the optimal performance. RandomForest requires the setting of two parameters, F representing the number of features used in random selection for building the decision trees and K representing the number of trees to grow. Using grid search, we found that the best choice for F is two, especially in the cases where $K > 10$. As K increases, the performance asymptotically improves, but for values above 30, the improvements are very minor. After tuning, average recall rate above 97.5% were achieved[1].

Finally, we measured computational complexity in terms of Central Processing Unit (CPU) time needed for classification, in order to provide a relative comparison between the algorithms in terms of battery usage requirements. As a result of our measurements, we are able to rank the classifiers from lowest to highest complexity as follows: SVM, ANN, LR, BN, DT, NB, IBk, LWL and KStar. CPU times were measured on a desktop PC not a smartphone, so the results only provide information about the relative complexity of the classifiers. The RandomForest algorithm, although it does not generate the simplest classifier in terms of computational cost,

[1] Note that the performance results for the model selection and parameter tuning portions of this paper are not directly comparable because the performances was measured with different test sets.

provides the best performance with reasonable complexity. SVM and ANN are very good in terms of testing complexity. SVM time complexity scales linearly with the number of features and support vectors and also depends on the type of kernel used. ANN scales with the number of neurons and the complexity of the network. RandomForest time complexity is affected mostly by the choice of parameter K. Our CPU measurement times were made with $K = 10$, and at this setting, RandomForest took about 3 times longer for testing compared to SVM. Lastly, according to our results, IBk, LWL, and KStar are very expensive at the testing phase (i.e. the inference phase). Their time complexity scales linearly with the amount of training data used. They are often referred to as "lazy classifiers" because they do the bulk of the computation, not during training, but during testing.

[P5] examines the feasibility of an ice-aware maritime route optimization algorithm and presents a novel method for such purposes. Our aim is to increase the safety and efficiency of maritime transport under icy conditions. Earlier works in this area mainly used numerical methods that could not guarantee global optimum solutions. Our proposed method combines several elements, including (1) a sea spatial model, (2) ship maneuverability model, (3) sea ice model, and (4) ship performance model. The sea spatial model is a rectilinear grid of points, masked by a boolean criteria—whether the depth of the sea is greater than a chosen threshold (normally the draught of the ship). The ship maneuverability model consisted of a set of neighbor grid points with respect to each grid point, defining edges between nodes in a graph. The purpose of this model is to discretize the number of directions in which the ship can travel, making the route optimization algorithm more computationally tractable. The sea ice model used, called HELMI, was developed by the Finnish Meteorological Institute. Lastly, the ship performance model was previously developed by co-authors as originally presented in [27].

Route optimization is performed using the A* algorithm, which is a graph-based search algorithm that uses a heuristic to guide (i.e. speed up) the search process. The heuristic we used was the Euclidean distance between the current search point and the desired destination, divided by the maximum speed of the ship over open water.

The main novelty in this research is the development of an intuitive cost function that takes into account ice conditions and available icebreaker assistance. From a context awareness perspective, the ice conditions and status of the icebreaker system are the main aspects of context utilized in this application. The focus of this paper, however, was not on detecting context but on applying context awareness in order to build up a context-aware navigation service. This research is also the first application of the A* algorithm to long-range maritime route optimization. We present example results based on the method, using the Baltic Sea as a case study. Generated routes are compared with historical routes under the same ice conditions to provide preliminary validation of the method. Further validation will require more extensive research, including field trials using the generated routes in real ice conditions.

5.2 Mapping of Publications to Research Areas

Figure 5.1 presents a mapping between the included publications [P1]-[P5] and different areas of research within the topic of context awareness. These areas can be divided into three broad areas according to the subject matter and methodology used: (1) background and literature review, (2) concepts and theory, and (3) different use case scenarios. Publications [P1] and [P2] fall primarily within the first and second areas, respectively. Publications [P3]-[P5], although containing some elements from the first two areas, mainly deal with different use case scenarios where context awareness can be applied.

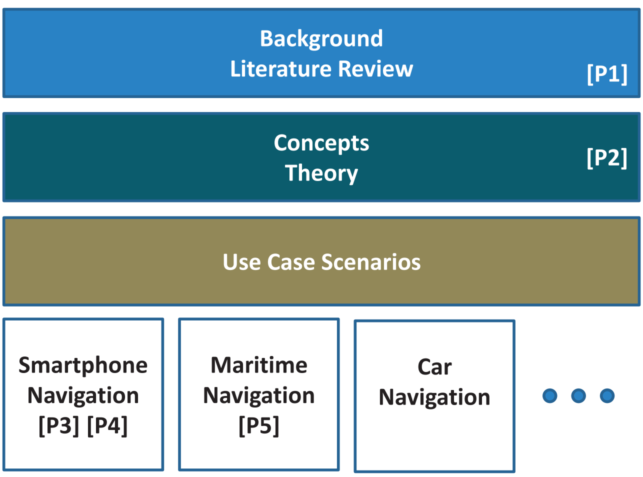


Fig. 5.1: *Mapping of included publications to research areas. Many use case scenarios have yet to be explored. Further discussion on future use case scenarios can be found in Section 6.4.2.*

As Figure 5.1 indicates, there are many potential use case scenarios that are not addressed in this thesis. Our original research plan was to cover as many different use case scenarios as possible, but due to time constraints and project limitations, only two separate use case scenarios could be investigated. Our plans to cover additional use case scenarios will be discussed briefly in Chapter 6.

5.3 Author's Contributions to the Publications

This section outlines the main contributions of the author of this thesis to the included publications.

[P1]: The thesis author was the main author of this chapter, whereas the co-author provided only editorial comments to a near-final draft. The thesis author conducted all the necessary background literature review and independently decided on the detailed contents of the chapter. The author also came up with the idea to use Hermagoras's "seven elements of circumstance" to organize and describe the different elements of context. The author also independently identified and collated the various Android Application Programming Interfaces (APIs) relevant to context awareness.

[P2]: The thesis author was the main author of this chapter, whereas the co-author provided only editorial comments to a near-final draft. Similar to the case of [P1], the thesis author conducted all the necessary background literature review and independently decided on the detailed contents of the chapter. The author created all of the figures in the chapter and originated the concept of the "context pyramid". Finally, one of the main contributions of this chapter is the formulation and description (including figures) of rather complex mathematical concepts (e.g. Support Vector Machines) in such a way that they are easily understandable for someone with basic knowledge in algebra and probability.

[P3]: The thesis author was the second author of this publication, but the first author has affirmed in writing that the thesis author's contribution was roughly equal to that of the first author. The thesis author contributed equally to the design and implementation of the experiments. Implementation of the data collection application on a smartphone was primarily the responsibility of the thesis author. He also assisted in the analysis of the test results. In addition, he was the originator of the idea used for indoor-outdoor detection based on GPS signal-to-noise ratio and WiFi signal strength and implemented this method on a smartphone. The thesis author contributed to the development of the WiFi fingerprinting indoor positioning system and prepared the test environment by measuring and marking reference points and setting up additional WiFi access points. As stated earlier, he was the originator of the “context pyramid” concept, which is also described and employed in this paper. He was also the originator of the idea of using a graph-based grouping of the reference points. Lastly, he contributed to the preparation of the manuscript, including writing some sections and editing it in its entirety.

[P4]: The thesis author was the sole author of this publication and received assistance only in data collection, as well as general guidance from his supervisors. He also implemented all the necessary software used in the experiments, apart from the Weka software platform used in the data analysis. Some extensions to Weka, in terms of automating analysis and integrating Weka with Matlab, were also implemented by the author.

[P5]: The thesis author was the first author of this publication and was the originator of the idea of using a graph-based approach and the A* algorithm for ice-aware route optimization. He formulated the concept of combining a sea spatial model, ship maneuverability model, sea ice model, and ship performance model. In particular, the ship maneuverability model, which defines the graph structure and discretizes the ship’s maneuverability, was the thesis author’s own invention. Furthermore, he implemented the optimization algorithm in Matlab, basing the implementation only roughly on an open source implementation. The cost function was developed mainly by the thesis author, although various preliminary ideas were discussed together with the second author. The parts of the method that were not developed or provided by the thesis author were the resistive ship speed model, the ice data, and the historical ship data, all of which were provided by co-authors. Finally, the thesis author led the manuscript preparation, writing most sections, preparing all figures, and editing the manuscript in its entirety.

6. CONCLUSIONS

This chapter offers some conclusions based on our research. It is organized as follows: Section 6.1 briefly summarizes the thesis. Section 6.2 outlines our main findings. Section 6.3 explains the significance or potential impact of the results. Section 6.4 describes our future work planned in the areas addressed by this thesis. Finally, Section 6.5 provides a few concluding remarks.

6.1 Summary

The overall goal of this thesis was to improve our understanding of how computing devices can better understand us and our needs. As argued in this thesis, such understanding is often embodied, at least partly, in a concept known as context awareness. The primary method used to endow computers with context awareness has been—and we argue it will continue to be—machine learning.

In examining these topics, we have narrowed the focus to application areas related to navigation. Despite this narrowing of application areas, there are still many diverse needs in navigation, and this thesis focused on three particular use cases within navigation where context awareness is deemed beneficial: (1) detecting of different human activities inside a typical office environment to improve indoor location tracking, (2) detecting different "mobility contexts" of a smartphone user to improve outdoor location tracking, and (3) enabling "ice aware" route optimization for ships sailing in ice-covered waters to improve and automate the route planning needs of such ships. These use cases demonstrate the breadth of potential application areas of context-aware technology. Three of the included publications ([P3]–[P5]) aim to improve the state-of-the-art in these application areas by introducing either novel methods, novel combinations of existing methods, or in-depth analysis of the performance of existing methods.

In addition to examining these application areas, this thesis has extensively reviewed the literature concerning context awareness and machine learning. In presenting and summarizing these topics, we have attempted to provide clear, tutorial-like examples, in order to aid readers unfamiliar with these subjects. We also provide a chapter on navigation to familiarize the reader with the intended application area.

In presenting the conceptual underpinnings of context awareness, we have introduced two conceptual frameworks for understanding context awareness and contextual reasoning. The first was adapted from the writings of an ancient Greek orator Hermagoras of Temnos, known as the "seven circumstances". The second, which we have dubbed the "context pyramid," describes the process of contextual reasoningin terms of six levels ranging from raw data to "rich context". These two frameworks, general in nature, can assist the researcher and developer aiming to build context-aware systems by dividing the problem up into different categories of contextual information and steps in contextual reasoning.

On the topic of machine learning, this thesis has examined the original goal of machine learning, as envisioned by pioneers such as Arthur Samuel. We emphasize the concept of *automatic learning* using computer chess as an example. We then examine the modern notion of machine learning, including the two major types, supervised learning and unsupervised learning. We provide a tutorial-like example of both types of learning, using an example problem from context awareness.

During this overview on machine learning, we have emphasized the importance and benefits of automatic learning. That is, supervised learning usually requires manual labeling of training data, whereas unsupervised learning can largely meet the desire for automated learning, although it often requires some human interpretation of the results.

Finally, the included publications provide further details on machine learning and its application to context awareness, and in particular [P3] and [P4] demonstrate the use of machine learning in problems related to navigation. Lastly, [P5] provides an example application of context awareness in the field of navigation, i.e. an ice-aware route optimization method.

6.2 Main Findings

In Section 1.2, we identified two overall research questions addressed by this thesis. These questions are implicitly discussed in various parts of the thesis, including the five publications. In this section, we explicitly summarize our findings regarding these questions.

What are the benefits and constraints of introducing context awareness in navigation?

The main benefit of introducing context awareness in navigation is to increase the level of automation that can be achieved in performing navigation functions. Although humans are inherently good at recognizing and understanding context, computers are relatively deficient in this ability. Nonetheless, the techniques described in this thesis and other state-of-the-art context awareness research help to endow computers with such abilities. Several examples have been given related to the three tasks investigated in this thesis, and their application to navigation has been described. Without these abilities, the user of a navigation system would have a greater burden in terms of explicitly switching into different navigation modes, or performing manual integration of data.

In the particular case of maritime navigation applications, awareness about ice conditions (as a function of space and time) can be exploited to perform automated route optimization. Such capability could augment or even replace the currently human-intensive task of route planning performed by crews of ships sailing in ice-covered waters. Our research showed that graph-based approaches are feasible for modeling maritime transportation in ice-covered waters and that the A* algorithm can be applied to find optimal paths. In order to realize an implementation of the A* algorithm, our research presents a simple but novel cost function that takes into account the operational constraints posed by ice breaker assistance. Essentially, this cost function captures contextual information about a ship's theoretical speed through an ice field, taking into account the ship's own ice-breaking performance and possible assistance from an ice breaker. The results of this method allow different proposed routes to be compared, in terms of voyage times, and provides decision support for the final selection of the ship's planned route.

The main constraint of context awareness in navigation is that the level of detail concerning context that can be recognized by computers is relatively low. Although the abilities in the state-of-the-art are rapidly increasing, still a human can much better describe the important elements of context in a succinct yet rich manner. Furthermore, the models developed for context awareness are not error free. In our research work, we achieved successful context recognition in the range of 90% to 98% of the studied cases. Efficient, fail-safe methods to deal with errors from context awareness systems must be developed in an application-specific manner. This topic has not been investigated in this thesis, and the literature on the subject remains scarse.

Another constraint of context awareness in navigation is that context awareness systems consume resources, in terms of the power, mass, and cost used for sensors, computational units, etc. If the improvement to the navigation system is only marginal, or if the user does not feel significantly burdened by the manual alternative to a context-aware system, then it will be difficult to justify the use of these extra resources. This research area is perhaps still to immature to determine with certainty whether the benefits associated with context awareness justify the costs, especially in the domain of navigation.

How can machine learning be used to build context or situation awareness, in order to solve problems in navigation?

This thesis, in particular [P3] and [P4], shows that machine learning is a powerful tool to enable context awareness, in order to solve various problems in navigation. We employ a number of different supervised learning algorithms, and in particular the method of building classification models via supervised learning suits the problem area very well. Although we did not compare the use of supervised learning against other types of model building, we can at least conclude that machine learning produces performance levels that are quite promising from a research standpoint. We have shown that smartphones can reliably detect different mobility contexts (>97% recall rate) and detect different office-environment activities (>90% recall rate). We have also introduced a method to detect whether a user is indoors or outdoors. Existing algorithms from supervised learning provide adequate levels of performance for these use cases, although generalization to large user populations will require the collection of more extensive training data. Especially in the case of context-aware smartphone applications, context awareness is presently feasible and can be realized using existing machine learning techniques.

Another point concerning the above research question concerns how machine learning techniques should be evaluated. Although machine learning constitutes a powerful set of methods for endowing computers with context awareness, a systematic evaluation of different available machine learning algorithms should be undertaken when applying machine learning to the problem of context awareness, especially if the aim is to maximize performance. This important fact is often overlooked by navigation researchers working on context awareness. After evaluating the performance of 20 different supervised learning algorithms, we determined that the RandomForest algorithm performed the best on our dataset. Our results also showed that performance is optimized only after applying extensive feature selection and parameter tuning. Another general recommendation from our research is that, due to the need to train many different classification models (with various feature sets and parameter settings), a high level of automation for this type of analysis is desirable, and we developed some software tools to improve the automation of this type of analysis.

One limitation of supervised learning is that the cost of obtaining labeled data is, in general, quite high. This issue will be discussed further in Section 6.4.3.

In this thesis, we have primarily investigated machine learning techniques that are built upon assumptions that the input data are independent and identically distributed. This is one major limitation of our research, since it is clear that most aspects of context are temporally correlated. Some machine learning techniques which exploit temporal correlations, such as HMMs are discussed in [P2], but thus far we have not applied these techniques in our context awareness research. This topic will be discussed further in Section 6.4.1.

In summary, our overall research on the use of machine learning in context awareness shows the feasibility of developing context-aware navigation applications for the three use-case scenarios investigated. In addition, our research suggests many other applications of context awareness are evident in emerging technologies related to navigation. We believe context awareness will play an even stronger role in navigation in the future, especially as so-called "smart devices" continue to proliferate.

6.3 Significance of the Results

This thesis contributes to the overall body of knowledge on context awareness, which as previously discussed, has many potential applications in the field of navigation. We have introduced two general and flexible frameworks related to context awareness—one for the systematic encoding of contextual information and the other for the processing of raw sensor data into "rich context". These frameworks serve as a methodological skeleton on which other researchers and developers can build new context-aware systems, not only in navigaion but more generally.

Much of this thesis focuses on context awareness that can be achieved using only the sensors in smartphones. Because of the widespread prevalence of smartphones in modern society, the results of smartphone-based context awareness research have a strong potential for widespread adoption. We can compare this to previous context awareness studies that rely on custom sensors being installed in the environment. By requiring installation of new hardware into the environment, the cost of adoption increases. According to our knowledge, our research is the first to look at enabling office-related context awareness using only smartphone sensors and standard WLAN access points.

Regarding the performance results reported in our mobility context research, it is difficult to determine definitively whether or not our results represent an improvement over the state-of-the-art. This is due to the fact that we have not compared our results against those obtained with comparable datasets. This issue will be further discussed below in Section 6.4.3. What we can conclude confidently is a methodological fact: that extensive analysis, consisting of evaluating different machine learning algorithms, performing feature selection, and systematically tuning parameters of the algorithm, will result in better performance regardless of the dataset[1]. Surprisingly, many research works in context awareness, especially in the navigation community, overlook this fact and report results from evaluating only one or a few machine learning algorithms and without reporting any results of parameter tuning. We hope that the methodology and results described in this thesis provides a new example as to the benefits of such detailed analysis.

Lastly, our research demonstrates the feasibility of developing an ice-aware maritime

[1] Admittedly, this is not a novel finding.

navigation system, specifically to provide automatic route optimization. Although further refinement is needed, the proposed methods have the potential to provide significant savings for the maritime transportation system. After presenting these methods to Director-General of the Finnish Ministry of Transportation and Communications, Mr. Pekka Plathan, he commented that this research has the potential to save billions for the Finnish maritime industry. Further innovations, improvements, and validation work, however, are required before such savings can be realized.

Ice-aware route optimization can not only bring economic benefits for the maritime industry, but it also has the potential to provide safety benefits. The methods described in this thesis are flexible in that the cost function can be modified to optimize non-economic factors as well. For example, using models for the risk that ice poses towards damaging ships or getting the ship completely stuck in the ice, one can design an appropriate cost function aimed to minimize these risks.

6.4 Future Work

In many ways, this thesis has only scratched the surface in exploring context awareness for navigation applications. In tackling the broader goal "to improve our understanding of how computing devices can better understand us and our needs," we feel even less compelled to declare our work complete. This section outlines some of our planned future work in developing context-aware navigation applications.

Our future work can be divided into three broad categories: (1) future work in the three application areas covered by the included publications, (2) future work in new application areas, and (3) future work that can benefit context awareness broadly. These areas of future work will be discussed in separate sub-sections below.

6.4.1 Future Work in Investigated Applications

In our research on detecting office-environment contexts, we investigated a small number of different workplace contexts, including working in one's office, having lunch, taking a break, and fetching coffee or water. There are obviously a large number of other workplace contexts that could be investigated, such as having a meeting, giving a presentation, having an impromptu conversation, talking on the phone, etc. In our research, we grouped all "non-defined" contexts into a single category called "undefined context". Also, we conducted this research in only one particular office environment. Extending this research to many diverse office environments and types of work would improve the robustness of the results.

Another way to improve this line of research would be to expand the sources of raw data to include other sensors. For example, smartphones have microphones, and sound could be an important source of contextual information. In fact, we informally explored using audio as a feature, but there are some challenges in this regard. For example, when the phone is in the pocket, the audio signal becomes very muffled and contact with clothing can cause loud undesirable signals. Nonetheless, we believe audio is an important source of contextual information, and we aim to explore this further in the future.

Also, different social context aspects of the workplace environment will be included in our future research. In this thesis, we did not address social context at any depth, but especially in an office environment, it should be feasible to recognize different social contexts because in many workplaces the identities of most of the people present are largely known. Using various proximity sensing technologies, a smartphone could apply contextual reasoning about different social contexts, such as "with the boss," "with subordinates," "with colleagues," "with a customer," etc. Such contextual information may not be needed necessarily for navigation applications, but it would certainly have other applications related to mobile computing.

On the subject of mobility contexts, we also plan to expand the range of contexts under investigation, such as cycling, riding trams, riding metros, etc. In addition, we plan to investigate whether we can recognize a number of other mobility-related leisure activities, such as hiking, berry or mushroom picking[2], playing golf, dog-walking, etc.

Also, in this thesis, for detecting mobility context we only utilized two smartphone sensor types, namely GPS and accelerometers. In the future, we aim to include other types of sensors, such as gyroscopes, pressure sensors, microphones, and light sensors.

Lastly, we plan to expand the number of test subjects participating in the collection

[2] These are popular leisure activities in Finland.

of training data. This is important to ensure that the trained models are robust. In this thesis, we asked the test subjects to keep the smartphone in a particular location (pants pocket), so in future research we will also study the effects of placement of the smartphone in different locations, such as a backpack, handbag, belt "holster," etc.

In both [P3] and [P4], all of the machine learning algorithms utilized operate independently on each data sample. That is, no time dependence between the data samples is exploited. As already mentioned in Section 6.2, contexts are strongly correlated with time, so we would expect to improve context recognition performance by using models that exploit these temporal correlations. Our future work will focus on such models and algorithms, including HMMs, Conditional Random Fields (CRF), and Markov chain Monte Carlo (MCMC) methods.

On the subject of ice-aware route optimization, further work is needed to validate the proposed method. This should include further analysis of historical data from Automatic Identification System (AIS), as well as simulator-based studies and actual testing of routes at sea. Also, in this thesis the route optimization method focused on minimizing travel times for ships, but in the future other aspects should be investigated, such as fuel usage, operational efficiency, safety, and reliability. Lastly, contextual information regarding the ice conditions should be enriched compared to the model used in this thesis. For example, the current model does not take into account ice compression, which can have a large effect on ice-going ship performance.

6.4.2 Future Applications

In Figure 5.1 we hinted at future application areas or "use case scenarios," many of which fall outside the domain of navigation. Several of the highlighted use case scenarios are part of near future work. For example, in one recently initiated project, we aim to develop a "tactical situation awareness system" for soldiers.

Military applications of context awareness are particularly promising because the cost limitations are not as strict as in other application areas and specially-designed sensors can be installed, e.g. attached to various body parts of a soldier (helmet, boots, chest, etc.) or to other military equipment, providing a rich set of raw sensor data from which to generate context awareness. On the other hand, in military applications, reliability requirements are very high, and typically there is a strong requirement for real-time functionality. For example, if a system is designed to detect when a soldier is in danger or injured, then false negatives, as well as false positives could prove very costly.

Another application area that has strong potential is healthcare and fitness monitoring. With the growing popularity of "wearable devices," such as smartwatches and small heart-rate monitors, such applications have greater widespread consumer appeal. Many devices already exist that can, e.g. monitor calorie usage by tracking steps, but it remains a challenge to reliably and automatically detect different activities such as walking, running, cycling, hiking, etc. This is, of course, strongly overlapping with the topic of [P4], but we believe healthcare and fitness monitoring can go much beyond mere "mobility context" and incorporate other aspects, such as recognizing social interactions, detecting abnormal health or changes to a person's routine that might affect health and fitness, and warning users of dangerous or unhealthy situations. The concept of a "personal health assistant" is not really a matter of science-fiction but could be realized in the coming years. Context awareness and machine learning are the technologies that are likely to make this concept a reality.

6.4.3 General Issues and Potential Solutions

Lastly, we have noticed in our research several general issues that are relevant to context awareness in a broad sense. These issues are summarized as follows:

1. Supervised learning requires labeled data, and labeled data are expensive.
2. There is a lack of standardization in context awareness research.
3. Many context awareness experiments are not easy to repeat or independently verify.

The first general issue above is related to the use of supervised learning, which is often the adopted approach in many research works (such as in [P3] and [P4]). While supervised learning has many advantages compared to unsupervised learning, it can be very costly and time-consuming to generate the required labeled data. Furthermore, it is generally the case that the more data that can be collected, the more performant and reliable the resulting model will be. For example, if we are aiming to develop a context-aware smartphone application that works well across a large population of users, then we will need to collect training data from a large, diverse population of test users. This is very costly, especially in a research setting.

There are two potential solutions to this issue. The first is that researchers and developers would publish and share their training data. This would benefit the overall research community. We have practiced this approach in publication [P4], and there are a few other examples of data sharing in the context awareness literature (e.g. [14] [119]). Generally, this is not a common practice in context awareness research. The second approach would be to collect a sizable amount of labeled training data, and then to supplement it with unlabeled data (which is less costly to collect). Performing machine learning using a combination of labeled and unlabeled data is known as *semi-supervised learning*. This topic is outside the scope of this thesis but will be explored in our future work. As an example of this approach, a research and development team could collect a limited amount of labeled data using its own staff and volunteers and then supplement it with a large amount of crowdsourced unlabeled data. This is exactly the approach we are taking in a recently initiated project called MyGeoTrust (see [120]). The other approach was discussed in Section 3.4; using unlabeled data to strategize about and prioritize the collection of labeled data.

The second general issue has to do with standardization. To put it precisely, there is a lack of standardization in context awareness research, and this issue makes it difficult to compare results among different studies. As described in Chapter 4, context is understood in many different ways, and there is no one "correct" way to categorize and organize the context space. Table 2.1 demonstrates this problem.

This lack of standardization is understandable, due to the fact that different researchers have different applications in mind and different ideas about how to segregate the context space, but it would be more beneficial for the overall research community if some level of standardization were applied. Many context ontologies have been proposed in the literature (e.g. [121] [122] [123] [124] [125]), and one of these could form the basis of a context ontology standard. Then, when presenting results, researchers could reference these standards, i.e. "the following classification results are according to standard X.Y...". Also, there is no reason to limit results to one particular standard; data could be processed, according to several different standards and presented in the same publication. The problem is that no forerunners for a standard have emerged, and no good software tools for working with the proposed ontologies have been made available (to our knowledge). In our future work we aim to contribute to and advance the notion of standard context ontologies, including open source tools for working with such ontologies.

The last general issue we would like to discuss is somewhat related to the second issue, and the solution is ironically similar to the first solution described above. One of the long-standing tenets of scientific research is reproducibility. Experiments should be described in enough detail so that other researchers can independently verify the results. In the case of context awareness research, this means that an independent researcher should be able repeat another researcher's data collection, apply the same algorithms, and achieve similar, if not identical, results. In reality, there are so many complex factors related to the environment, devices, and test subjects that collecting comparable data that produces comparable results is not always realistic.

The solution is straight-forward. As an alternative, context researchers should always publish the data upon which their results are based, along with sufficient documentation so that the data is usable by independent researchers. As already stated, this is rarely done in context awareness research. It is, however, a common practice in the machine learning community to test techniques against benchmark data. For example, the University of California, Irvine (UCI) maintains a repository of over 300 datasets that can be used for machine learning research [126]. Unfortunately, very few of these datasets relate to context awareness. A few other sources of open data for context awareness research exist, including the Mobile Data Challenge (MDC) Dataset collected as part of the Lausanne Data Collection Campaign [127] and data from the University of Helsinki's "Context project" [128]. We aim to follow open data practices in our future work and also to actively promote this practice, either by promoting the use of UCI's machine learning repository or by setting up a dedicated portal for context awareness research.

6.5 Concluding Remarks

The remainder of this thesis consists of reprints of the five included publications described earlier. The order of the publications has been chosen to go from the most general to more specific and detailed applications. They can be read, however, in any order, depending on the reader's specific interests.

BIBLIOGRAPHY

[1] T. Mitchell, B. Buchanan, G. DeJong, T. Dietterich, P. Rosenbloom, and A. Waibel, "Machine learning," *Annual Review of Computer Science*, vol. 4, pp. 417–433, 1990.

[2] E. Brynjolfsson and A. McAfee, *The second machine age: work, progress, and prosperity in a time of brilliant technologies*. New York: W. W. Norton & Company, 2014.

[3] Canalys.com, Ltd., "64 million smart phones shipped worldwide in 2006." Available online: http://bit.ly/1HSNZBZ, 2007.

[4] Gartner, "Gartner says worldwide smartphone sales reached its lowest growth rate with 3.7 per cent increase in fourth quarter of 2008," March 2009.

[5] N. Mawston, "Worldwide smartphone population tops 1 billion in Q3 2012." Available online: http://bit.ly/1DZuyYo, October 2012. (archived on 26 February 2015).

[6] "Smartphone users expected to hit 2.5 billion next year." *The Korea Times*, Available online: http://bit.ly/1HSO5cQ, 2014.

[7] J. McCarthy, M. Minsky, N. Rochester, and C. Shannon, "A proposal for the Dartmouth summer research project on artificial intelligence," *AI Magazine*, 2006. Reprint of proposal originally written in 1955.

[8] D. Crevier, *AI: The tumultuous history of the search for artificial intelligence*. Basic Books, Inc., 1993.

[9] P. D. Groves, L. Wang, D. Walter, H. Martin, K. Voutsis, and Z. Jiang, "The four key challenges of advanced multisensor navigation and positioning," in

Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION, pp. 773–792, IEEE, 2014.

[10] S. Abolfazli, Z. Sanaei, A. Gani, F. Xia, and L. T. Yang, "Rich mobile applications: genesis, taxonomy, and open issues," *Journal of Network and Computer Applications*, vol. 40, pp. 345–362, 2014.

[11] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems*. Artech house, 2013.

[12] M. Petovello, "How does a GNSS receiver estimate velocity?." Available online: http://bit.ly/1GwTuJN, March/April 2015. Inside GNSS.

[13] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, "Transportation mode detection using mobile phones and GIS information," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 54–63, November 2011.

[14] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang, "Big data small footprint: The design of a low-power classifier for detecting transportation modes," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1429–1440, 2014.

[15] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Proceedings of the national conference on artificial intelligence*, vol. 20, pp. 1541–1546, July 2005.

[16] K. Sankaran, M. Zhu, X. F. Guo, A. L. Ananda, M. C. Chan, and L.-S. Peh, "Using mobile phone barometer for low-power transportation context detection," in *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pp. 191–205, ACM, 2014.

[17] H. Yan and T. Selker, "Context-aware office assistant," in *Proceedings of the 5th international conference on Intelligent user interfaces*, pp. 276–279, ACM, 2000.

[18] M. Danninger and R. Stiefelhagen, "A context-aware virtual secretary in a smart office environment," in *Proceedings of the 16th ACM international conference on Multimedia*, pp. 529–538, ACM, 2008.

[19] H. W. Gellersen, A. Schmidt, and M. Beigl, "Multi-sensor context-awareness in mobile devices and smart artifacts," *Mobile Networks and Applications*, vol. 7, no. 5, pp. 341–351, 2002.

[20] P. Nurmi and P. Floréen, "Reasoning in context-aware systems: a position paper." Available online: http://www.cs.helsinki.fi/u/ptnurmi/papers/positionpaper.pdf, 2004.

[21] M. Tähti, V.-M. Rautio, and L. Arhippainen, "Utilizing context-awareness in office-type working life," in *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, pp. 79–84, ACM, 2004.

[22] Y. Manabe, H. Saito, K. Akiyama, R. Ikeda, S. Kanda, and K. Sugawara, "Perceptual functions for context-awareness of an office worker," in *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*, pp. 583–589, IEEE, 2010.

[23] L. M. Marti, R. Mayor, and S. M. Ma, "Managing states of location determination." US Patent Application 13/715,710, 19 June 2014.

[24] R. E. Guinness, "Beyond where to how: A machine learning approach for sensing mobility contexts using smartphone sensors," in *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013)*, (Nashville, TN), pp. 2868–2879, September 2013.

[25] J.-H. Nam, I. Park, H. J. Lee, M. O. Kwon, K. Choi, and Y.-K. Seo, "Simulation of optimal arctic routes using a numerical sea ice model based on an ice-coupled ocean circulation method," *International Journal of Naval Architecture and Ocean Engineering*, vol. 5, no. 2, pp. 210–226, 2013.

[26] M. Choi, H. Chung, H. Yamaguchi, and K. Nagakawa, "Arctic sea route path planning based on an uncertain ice prediction model," *Cold Regions Science and Technology*, vol. 109, pp. 61–69, 2015.

[27] V. Kotovirta, R. Jalonen, L. Axell, K. Riska, and R. Berglund, "A system for route optimization in ice-covered waters," *Cold Regions Science and Technology*, vol. 55, no. 1, pp. 52–62, 2009.

[28] C. R. Colon, "An Efficient GPS Position Determination Algorithm." *Air Force Institute of Technology*, Available online: http://1.usa.gov/1KFZnPK, 1999.

[29] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.

[30] R. Chen and R. E. Guinness, *Geospatial computing in mobile devices*. Boston: Artech House, 2014.

[31] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements and Performance Second Edition*. Lincoln, MA: Ganga-Jamuna Press, 2006.

[32] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1028–1035, 2009.

[33] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management science*, vol. 6, no. 1, pp. 80–91, 1959.

[34] "Luciad - geospatial situational awareness." *Company website:* http://www.luciad.com/, 2015.

[35] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[36] A. Turing, *Chess*, ch. The Essential Turing, Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life plus The Secrets of Enigma. Oxford University Press, 2004. Reprint of paper originally printed in 1953.

[37] C. E. Shannon, "XXII. programming a computer for playing chess," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 41, no. 314, pp. 256–275, 1950.

[38] F.-H. Hsu, *Behind Deep Blue: Building the computer that defeated the world chess champion*. Princeton University Press, 2002.

[39] D. Spicer and K. Tashev, "The quest to build a thinking machine: A history of computer chess," *CompuComputer: New exhibit showcases game's past and museum's future*, May 2006. A publication of the Computer History Museum.

[40] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, New Jersey: Prentice Hall, 3rd edition ed., 2010.

[41] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.

[42] K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[43] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[44] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[45] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[46] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[47] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *SIGMOD Rec.*, vol. 28, pp. 49–60, June 1999.

[48] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[49] J. A. Bilmes *et al.*, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models," *International Computer Science Institute*, vol. 4, no. 510, p. 126, 1998.

[50] N. Vlassis and A. Likas, "A greedy EM algorithm for gaussian mixture learning," *Neural processing letters*, vol. 15, no. 1, pp. 77–87, 2002.

[51] P. Smyth, "Note set 4: Finite mixture models and the EM algorithm." Lecture notes for course on Probabilistic Learning at University of California–Irvine, Available online: http://www.ics.uci.edu/ smyth/courses/cs274/notes/notes4.pdf, 2015.

[52] P. W. Foltz, W. Kintsch, and T. K. Landauer, "The measurement of textual coherence with latent semantic analysis," *Discourse Processes*, vol. 25, no. 2, pp. 285–307, 1998.

[53] M. Bazire and P. Brézillon, "Understanding context before using it," in *Modeling and Using Context: Lecture Notes in Computer Science* (A. Dey, B. Kokinov, D. Leake, and R. Turner, eds.), Springer Berlin/Heidelberg, 2005.

[54] J. McCarthy, "Notes on formalizing context," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, (San Mateo, California), pp. 555–562, Morgan Kaufmann, 1993.

[55] "Context." *Merriam-Webster Dictionary*, Available online: http://bit.ly/1LK7K2O, 2015.

[56] V. Akman and M. Surav, "Steps toward formalizing context," *AI magazine*, vol. 17, no. 3, p. 55, 1996.

[57] Wikipedia, "Five Ws — Wikipedia, the free encyclopedia." Available online: http://bit.ly/1SIXRkI, 2015.

[58] B. S. Bennett, "Hermagoras of temnos," *Classical Rhetorics and Rhetoricians: Critical Studies and Sources*, p. 187, 2005.

[59] J.-y. Hong, E.-h. Suh, and S.-J. Kim, "Context-aware systems: A literature review and classification," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509–8522, 2009.

[60] B. Schilit and M. Theimer, "Disseminating active map information to mobile hosts," *Network, IEEE*, vol. 8, no. 5, pp. 22–32, 1994.

[61] M. Weiser, "The computer for the 21st century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[62] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.

[63] T. Moran and P. Dourish, "Introduction to this special issue on context-aware computing," *Human-Computer Interaction*, vol. 16, no. 2-4, pp. 87–95, 2001.

[64] J. McCarthy, *Programs with common sense*. Defense Technical Information Center, 1963.

[65] J. McCarthy, "Generality in artificial intelligence," *Communications of the ACM*, vol. 30, no. 12, pp. 1030–1035, 1987.

[66] D. Foxvog, "Cyc," in *Theory and Applications of Ontology: Computer Applications*, pp. 259–278, Springer, 2010.

[67] R. V. Guha, *Contexts: a formalization and some applications*, vol. 101. Stanford University Stanford, CA, 1991.

[68] J. McCarthy and S. Buvac, "Formalizing context (expanded notes)," in *Computing Natural Language, CSLI, Lecture Notes* (A. Aliseda, R. J. van Glabbeek, and W. D., eds.), pp. 13–50, Center for the Study of Language and Information, Stanford University, 1998.

[69] S. Buvač and I. A. Mason, "Propositional logic of context," in *Proceedings of the eleventh national conference on artificial intelligence*, sn, 1993.

[70] P. Brézillon, "Context in artificial intelligence: I. a survey of the literature," *Computers and artificial intelligence*, vol. 18, pp. 321–340, 1999.

[71] V. Akman, "Context in artificial intelligence: a fleeting overview," in *La svolta contestuale* (C. Penco and V. Akman, eds.), Milano: McGraw-Hill, 2002.

[72] M. Tähti, V.-M. Rautio, and L. Arhippainen, "Utilizing context-awareness in office-type working life," in *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, pp. 79–84, ACM, 2004.

[73] M. Heiskala, E. Palomäki, M. Vartiainen, K. Hakkarainen, and H. Muukkonen, "A research framework for the smartphone-based contextual study of mobile knowledge work," in *Design, User Experience, and Usability. User Experience Design for Diverse Interaction Platforms and Environments*, pp. 246–257, Springer, 2014.

[74] X. Huang, J. Weng, and Z. Zhang, "Office presence detection using multimodal context information," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 3, pp. iii–773, IEEE, 2004.

[75] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, IEEE, 1999.

[76] Y. Manabe, H. Saito, K. Akiyama, R. Ikeda, S. Kanda, and K. Sugawara, "Perceptual functions for context-awareness of an office worker," in *Cognitive Informatics (ICCI), 2010 9th IEEE International Conference on*, pp. 583–589, IEEE, 2010.

[77] K. Kiyokawa, M. Hatanaka, K. Hosoda, M. Okada, H. Shigeta, Y. Ishihara, F. Ooshita, H. Kakugawa, S. Kurihara, and K. Moriyama, "Owens luis–a context-aware multi-modal smart office chair in an ambient environment," in *Virtual Reality Short Papers and Posters (VRW), 2012 IEEE*, pp. 1–4, IEEE, 2012.

[78] K. K. Rachuri, C. Efstratiou, I. Leontiadis, C. Mascolo, and P. J. Rentfrow, "Smartphone sensing offloading for efficiently supporting social sensing applications," *Pervasive and Mobile Computing*, vol. 10, pp. 3–21, 2014.

[79] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor activity context detection for wearable computing," in *Ambient Intelligence*, pp. 220–232, Springer, 2003.

[80] S. Pirttikangas, K. Fujinami, and T. Nakajima, "Feature selection and activity recognition from wearable sensors," in *Ubiquitous Computing Systems*, pp. 516–527, Springer, 2006.

[81] H. Park, J. Park, H. Kim, J. Jun, S. Hyuk Son, T. Park, and J. Ko, "Relisce: Utilizing resource-limited sensors for office activity context extraction," *Systems, Man, and Cybernetics: Systems, IEEE Transactions on*, vol. 45, pp. 1151–1164, Aug 2015.

[82] M. J. Duncan, H. M. Badland, and W. K. Mummery, "Applying GPS to enhance understanding of transport-related physical activity," *Journal of Science and Medicine in Sport*, vol. 12, no. 5, pp. 549–556, 2009.

[83] S. A. Hoseini-Tabatabaei, A. Gluhak, and R. Tafazolli, "A survey on smartphone-based systems for opportunistic user context recognition," *ACM Computing Surveys (CSUR)*, vol. 45, no. 3, p. 27, 2013.

[84] V. Pejovic and M. Musolesi, "Anticipatory mobile computing: A survey of the state of the art and research challenges," *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, p. 47, 2015.

[85] Y. Kwon, K. Kang, and C. Bae, "Unsupervised learning for human activity recognition using smartphone sensors," *Expert Systems with Applications*, vol. 41, no. 14, pp. 6067–6074, 2014.

[86] F. Foerster and J. Fahrenberg, "Motion pattern and posture: correctly assessed by calibrated accelerometers," *Behavior research methods, instruments, & computers*, vol. 32, no. 3, pp. 450–457, 2000.

[87] S.-W. Lee and K. Mase, "Activity and location recognition using wearable sensors," *IEEE pervasive computing*, vol. 1, no. 3, pp. 24–32, 2002.

[88] J. Lester, T. Choudhury, N. Kern, G. Borriello, and B. Hannaford, "A hybrid discriminative/generative approach for modeling human activities," in *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 766–772, 2005.

[89] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 10, no. 1, pp. 119–128, 2006.

[90] J. Suutala, S. Pirttikangas, and J. Röning, "Discriminative temporal smoothing for activity recognition from wearable sensors," in *Ubiquitous Computing Systems* (H. Ichikawa, W.-D. Cho, I. Satoh, and H. Youn, eds.), vol. 4836 of *Lecture Notes in Computer Science*, pp. 182–195, Springer Berlin Heidelberg, 2007.

[91] K. Kunze and P. Lukowicz, "Dealing with sensor displacement in motion-based onbody activity recognition systems," in *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 20–29, ACM, 2008.

[92] J. Yin, Q. Yang, and J. Pan, "Sensor-based abnormal human-activity detection," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 8, pp. 1082–1090, 2008.

[93] J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics*, pp. 1–10, ACM, 2009.

[94] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, pp. 13:1–13:23, 2010.

[95] L. Pei, R. Chen, J. Liu, W. Chen, H. Kuusniemi, T. Tenhunen, T. Kröger, Y. Chen, H. Leppäkoski, and J. Takala, "Motion recognition assisted indoor wireless navigation on a mobile phone," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010)*, pp. 3366–3375, 2010.

[96] K. Frank, M. Nadales, P. Robertson, and M. Angermann, "Reliable real-time recognition of motion related human activities using MEMS inertial sensors," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2010)*, pp. 2919–2932, 2001.

[97] L. Pei, R. Chen, J. Liu, H. Kuusniemi, Y. Chen, T., and Tenhunen, "Using motion-awareness for the 3d indoor personal navigation on a smartphone," in *Proceedings of the 24th International Technical Meeting of The Institute of Navigation (ION GNSS 2011), Portland, Oregon, USA, September 20-23*, 2011.

[98] M. Susi, D. Borio, and G. Lachapelle, "Accelerometer signal features and classification algorithms for positioning applications," in *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation*, pp. 158–169, 2011.

[99] J. B. Bancroft, D. Garrett, and G. Lachapelle, "Activity and environment classification using foot mounted navigation sensors," in *Proceedings of the 2012 Indoor Positioning and Indoor Navigation (IPIN) Conference*, 2012.

[100] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly

support vector machine," in *Ambient assisted living and home care*, pp. 216–223, Springer, 2012.

[101] M. Susi, V. Renaudin, and G. Lachapelle, "Motion mode recognition and step detection algorithms for mobile phone users," *Sensors*, vol. 13, no. 2, pp. 1539–1562, 2013.

[102] T. Feng and H. J. Timmermans, "Transportation mode recognition using gps and accelerometer data," *Transportation Research Part C: Emerging Technologies*, vol. 37, pp. 118–130, 2013.

[103] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems*, p. 13, ACM, 2013.

[104] L. Stenneth, *Detecting Human Activities Using Smartphones and Maps*. Ph.D. thesis, University of Illinois at Chicago, February 2014.

[105] H. Xia, Y. Qiao, J. Jian, and Y. Chang, "Using smart phone sensors to detect transportation modes," *Sensors*, vol. 14, no. 11, pp. 20843–20865, 2014.

[106] M. Elhoushi, J. Georgy, M. Korenberg, and A. Noureldin, "Robust motion mode recognition for portable navigation independent on device usage," in *Position, Location and Navigation Symposium-PLANS 2014, 2014 IEEE/ION*, pp. 158–163, IEEE, 2014.

[107] J. Parviainen, J. Bojja, J. Collin, J. Leppänen, and A. Eronen, "Adaptive activity and environment recognition for mobile phones," *Sensors*, vol. 14, no. 11, pp. 20753–20778, 2014.

[108] J.-H. Chiang, P.-C. Yang, and H. Tu, "Pattern analysis in daily physical activity data for personal health management," *Pervasive and Mobile Computing*, vol. 13, pp. 13–25, 2014.

[109] J.-M. Yu and S.-B. Cho, "A low-power context-aware system for smartphone using hierarchical modular Bayesian networks," in *Hybrid Artificial Intelligent Systems*, pp. 543–554, Springer, 2015.

[110] M. Choi, H. Chung, H. Yamaguchi, and L. W. A. De Silva, "Application of genetic algorithm to ship route optimization in ice navigation," in *Proceedings*

of the International Conference on Port and Ocean Engineering Under Arctic Conditions, 2013.

[111] I. H. Park, J. H. Nam, and K. S. Choi, "A graphical approach to determine the optimal sea route of icebreakers in arctic region," in *Proceedings of the Korean Association of ocean science and technology societies join conference*, June 2011. Published in Korean.

[112] J. Esa, "Fuel and economic efficiency of an ice-going vessel on the northern sea route," Master of Science thesis, Aalto University, June 2015.

[113] J. Montewka, F. Goerlandt, P. Kujala, and M. Lensu, "Towards probabilistic models for the prediction of a ship performance in dynamic ice," *Cold Regions Science and Technology*, vol. 112, pp. 14–28, 2015.

[114] D. LaPrairie, M. Wilhelmson, and K. Riska, *A transit simulation model for ships in Baltic ice conditions : Documentation of the calculation routine*. Helsinki University of Technology, 1995.

[115] P. Valanto, S. J. Jones, E. Enkvist, and K. Izumiyama, "The resistance of ships in level ice," *Transactions of the Society of Naval Architects and Marine Engineers*, vol. 109, pp. 53–83, 2001.

[116] B. H. Fock, A. Beitsch, D. Broehan, M. Dobynin, A. Gierisch, L. Kaleschke, T. Pohlmann, and K. H. Schlünzen, "Ice forecast and route optimization," 2012. Poster presented at the SMOSIce User Workshop, Hamburg, Germany.

[117] M. Dobrynin, B. H. Fock, A. M. Gierisch, T. Pohlmann, L. Kaleschke, H. Schlünzen, *et al.*, "Prediction of arctic sea ice for ship routing: Forecast experiment and ship cruise," in *OTC Arctic Technology Conference*, Offshore Technology Conference, 2015.

[118] A. K. Dey and G. D. Abowd, "Towards a better understanding of context and context-awareness," Tech. Rep. GIT-GVU-99-22, Georgia Institute of Technology, 1999.

[119] F. J. Ordónez, P. de Toledo, and A. Sanchis, "Activity recognition using hybrid generative/discriminative models on home environments using binary sensors," *Sensors*, vol. 13, no. 5, pp. 5460–5477, 2013.

[120] R. E. Guinness, H. Kuusniemi, J. Vallet, T. Sarjakoski, J. Oksanen, M. Islam, M. Syeed, H.-M. Halkosaari, P. Kettunen, M. Laakso, and M. Rönneberg, "MyGeoTrust: A platform for trusted crowdsourced geospatial data," in *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, (Tampa, Florida), 2015.

[121] H. Guermah, T. Fissaa, H. Hafiddi, M. Nassar, and A. Kriouile, "An ontology oriented architecture for context aware services adaptation," *arXiv preprint arXiv:1404.3280*, 2014.

[122] H. Chen, T. Finin, and A. Joshi, "An ontology for context-aware pervasive computing environments," *The Knowledge Engineering Review*, vol. 18, no. 03, pp. 197–207, 2003.

[123] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, "Ontology based context modeling and reasoning using OWL," in *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pp. 18–22, Ieee, 2004.

[124] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang, "An ontology-based context model in intelligent environments," in *Proceedings of communication networks and distributed systems modeling and simulation conference*, vol. 2004, pp. 270–275, 2004.

[125] T. Strang, C. Linnhoff-Popien, and K. Frank, "CoOL: A context ontology language to enable contextual interoperability," in *Distributed applications and interoperable systems*, pp. 236–247, Springer, 2003.

[126] M. Lichman, "UCI machine learning repository." Available online: http://archive.ics.uci.edu/ml/, 2015.

[127] J. K. Laurila, D. Gatica-Perez, I. Aad, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, "The mobile data challenge: Big data for mobile computing research," in *Pervasive Computing*, no. EPFL-CONF-192489, 2012.

[128] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen, "ContextPhone:

a prototyping platform for context-aware mobile applications," *Pervasive Computing, IEEE*, vol. 4, pp. 51–59, Jan 2005.

Suomen Geodeettisen laitoksen julkaisut:
Veröffentlichungen des Finnischen Geodätischen Institutes:
Publications of the Finnish Geodetic Institute:

1. Y. VÄISÄLÄ: Tafeln für geodätische Berechnungen nach den Erddimensionen von Hayford. Helsinki 1923. 30 S.
2. Y. VÄISÄLÄ: Die Anwendung der Lichtinterferenz zu Längenmessungen auf grösseren Distanzen. Helsinki 1923. 22 S.
3. ILMARI BONSDORFF, Y. LEINBERG, W. HEISKANEN: Die Beobachtungsergebnisse der südfinnischen Triangulation in den Jahren 1920-1923. Helsinki 1924. 235 S.
4. W. HEISKANEN: Untersuchungen über Schwerkraft und Isostasie. Helsinki 1924. 96 S. 1 Karte.
5. W. HEISKANEN: Schwerkraft und isostatische Kompensation in Norwegen. Helsinki 1926. 33 S. 1 Karte.
6. W. HEISKANEN: Die Erddimensionen nach den europäischen Gradmessungen. Helsinki 1926. 26 S.
7. ILMARI BONSDORFF, V.R. ÖLANDER, Y. LEINBERG: Die Beobachtungsergebnisse der südfinnischen Triangulation in den Jahren 1924-1926. Helsinki 1927. 164 S. 1 Karte.
8. V.R. ÖLANDER: Ausgleichung einer Dreieckskette mit Laplaceschen Punkten. Helsinki 1927. 49 S. 1 Karte.
9. U. PESONEN: Relative Bestimmungen der Schwerkraft auf den Dreieckspunkten der südfinnischen Triangulation in den Jahren 1924-1925. Helsinki 1927. 129 S.
10. ILMARI BONSDORFF: Das Theorem von Clairaut und die Massenverteilung im Erdinnern. Helsinki 1929. 10 S.
11. ILMARI BONSDORFF, V.R. ÖLANDER, W. HEISKANEN, U. PESONEN: Die Beobachtungsergebnisse der Triangulationen in den Jahren 1926-1928. Helsinki 1929. 139 S. 1 Karte.
12. W. HEISKANEN: Über die Elliptizität des Erdäquators. Helsinki 1929. 18 S.
13. U. PESONEN: Relative Bestimmungen der Schwerkraft in Finnland in den Jahren 1926-1929. Helsinki 1930. 168 S. 1 Karte.
14. Y. VÄISÄLÄ: Anwendung der Lichtinterferenz bei Basismessungen. Helsinki 1930. 47 S.
15. M. FRANSSILA: Der Einfluss der den Pendel umgebenden Luft auf die Schwingungszeit beim v. Sterneckschen Pendelapparat. Helsinki 1931. 23 S.
16. Y. LEINBERG: Ergebnisse der astronomischen Ortsbestimmungen auf den finnischen Dreieckspunkten. Helsinki 1931. 162 S.
17. V.R. ÖLANDER: Über die Beziehung zwischen Lotabweichungen und Schwereanomalien sowie über das Lotabweichungssystem in Süd-Finnland. Helsinki 1931. 23 S.
18. PENTTI KALAJA, UUNO PESONEN, V.R. ÖLANDER, Y. LEINBERG: Beobachtungsergebnisse. Helsinki 1933. 240 S. 1 Karte.
19. R.A. HIRVONEN: The continental undulations of the geoid. Helsinki 1934. 89 pages. 1 map.
20. ILMARI BONSDORFF: Die Länge der Versuchsbasis von Helsinki und Längenveränderungen der Invardrähte 634-637. Helsinki 1934. 41 S.
21. V.R. ÖLANDER: Zwei Ausgleichungen des grossen südfinnischen Dreieckskranzes. Helsinki 1935. 66 S. 1 Karte.
22. U. PESONEN, V.R. ÖLANDER: Beobachtungsergebnisse. Winkelmessungen in den Jahren 1932-1935. Helsinki 1936. 148 S. 1 Karte.
23. R.A. HIRVONEN: Relative Bestimmungen der Schwerkraft in Finnland in den Jahren 1931, 1933 und 1935. Helsinki 1937. 151 S.
24. R.A. HIRVONEN: Bestimmung des Schwereunterschiedes Helsinki-Potsdam im Jahre 1935 und Katalog der finnischen Schwerestationen. Helsinki 1937. 36 S. 1 Karte.
25. T.J. KUKKAMÄKI: Über die nivellitische Refraktion. Helsinki 1938. 48 S.
26. Finnisches Geodätisches Institut 1918-1938. Helsinki 1939. 126 S. 2 Karten.
27. T.J. KUKKAMÄKI: Formeln und Tabellen zur Berechnung der nivellitischen Refraktion. Helsinki 1939. 18 S.
28. T.J. KUKKAMÄKI: Verbesserung der horizontalen Winkelmessungen wegen der Seitenrefraktion. Helsinki 1939. 18 S.
29. ILMARI BONSDORFF: Ergebnisse der astronomischen Ortsbestimmungen im Jahre 1933. Helsinki 1939. 47 S.
30. T. HONKASALO: Relative Bestimmungen der Schwerkraft in Finnland im Jahre 1937. Helsinki 1941. 78 S.
31. PENTTI KALAJA: Die Grundlinienmessungen des Geodätischen Institutes in den Jahren 1933-1939 nebst Untersuchungen über die Verwendung der Invardrähte. Helsinki 1942. 149 S.
32. U. PESONEN, V.R. ÖLANDER: Beobachtungsergebnisse. Winkelmessungen in den Jahren 1936-1940. Helsinki 1942. 165 S. 1 Karte.
33. PENTTI KALAJA: Astronomische Ortsbestimmungen in den Jahren 1935-1938. Helsinki 1944. 142 S.
34. V.R. ÖLANDER: Astronomische Azimutbestimmungen auf den Dreieckspunkten in den Jahren 1932-1938; Lotabweichungen und Geoidhöhen. Helsinki 1944. 107 S. 1 Karte.
35. U. PESONEN: Beobachtungsergebnisse. Winkelmessungen in den Jahren 1940-1947. Helsinki 1948. 165 S. 1 Karte.
36. Professori Ilmari Bonsdorffille hänen 70-vuotispäivänään omistettu juhlajulkaisu. Publication dedicated to Ilmari Bonsdorff on the occasion of his 70th anniversary. Helsinki 1949. 262 pages. 13 maps.
37. TAUNO HONKASALO: Measuring of the 864 m-long Nummela standard base line with the Väisälä light interference comparator and some investigations into invar wires. Helsinki 1950. 88 pages.
38. V.R. ÖLANDER: On the geoid in the Baltic area and the orientation of the Baltic Ring. Helsinki 1950. 26 pages.
39. W. HEISKANEN: On the world geodetic system. Helsinki 1951. 25 pages.
40. R.A. HIRVONEN: The motions of Moon and Sun at the solar eclipse of 1947 May 20th. Helsinki 1951. 36 pages.
41. PENTTI KALAJA: Catalogue of star pairs for northern latitudes from 55° to 70°for astronomic determination of latitudes by the Horrebow-Talcott method. Helsinki 1952. 191 pages.
42. ERKKI KÄÄRIÄINEN: On the recent uplift of the Earth's crust in Finland. Helsinki 1953. 106 pages. 1 map.
43. PENTTI KALAJA: Astronomische Ortsbestimmungen in den Jahren 1946-1948. Helsinki 1953. 146 S.
44. T.J. KUKKAMÄKI, R.A. HIRVONEN: The Finnish solar eclipse expeditions to the Gold Coast and Brazil 1947. Helsinki 1954. 71 pages.
45. JORMA KORHONEN: Einige Untersuchungen über die Einwirkung der Abrundungsfehler bei Gross-Ausgleichungen. Neu-Ausgleichung des südfinnischen Dreieckskranzes. Helsinki 1954. 138 S. 3 Karten.

46. Professori Weikko A. Heiskaselle hänen 60-vuotispäivänään omistettu juhlajulkaisu. Publication dedicated to Weikko A. Heiskanen on the occasion of his 60th anniversary. Helsinki 1955. 214 pages.
47. Y. VÄISÄLÄ: Bemerkungen zur Methode der Basismessung mit Hilfe der Lichtinterferenz. Helsinki 1955. 12 S.
48. U. PESONEN, TAUNO HONKASALO: Beobachtungsergebnisse der finnischen Triangulationen in den Jahren 1947-1952. Helsinki 1957. 91 S.
49. PENTTI KALAJA: Die Zeiten von Sonnenschein, Dämmerung und Dunkelheit in verschiedenen Breiten. Helsinki 1958. 63 S.
50. V.R. ÖLANDER: Astronomische Azimutbestimmungen auf den Dreieckspunkten in den Jahren 1938-1952. Helsinki 1958. 90 S. 1 Karte.
51. JORMA KORHONEN, V.R. ÖLANDER, ERKKI HYTÖNEN: The results of the base extension nets of the Finnish primary triangulation. Helsinki 1959. 57 pages. 5 appendices. 1 map.
52. V.R. ÖLANDER: Vergleichende Azimutbeobachtungen mit vier Instrumenten. Helsinki 1960. 48 pages.
53. Y. VÄISÄLÄ, L. OTERMA: Anwendung der astronomischen Triangulationsmethode. Helsinki 1960. 18 S.
54. V.R. ÖLANDER: Astronomical azimuth determinations on trigonometrical stations in the years 1955-1959. Helsinki 1961. 15 pages.
55. TAUNO HONKASALO: Gravity survey of Finland in years 1945-1960. Helsinki 1962. 35 pages. 3 maps.
56. ERKKI HYTÖNEN: Beobachtungsergebnisse der finnischen Triangulationen in den Jahren 1953-1962. Helsinki 1963. 59 S.
57. ERKKI KÄÄRIÄINEN: Suomen toisen tarkkavaaituksen kiintopisteluettelo I. Bench mark list I of the Second Levelling of Finland. Helsinki 1963. 164 pages. 2 maps.
58. ERKKI HYTÖNEN: Beobachtungsergebnisse der finnischen Triangulationen in den Jahren 1961-1962. Helsinki 1963. 32 S.
59. AIMO KIVINIEMI: The first order gravity net of Finland. Helsinki 1964. 45 pages.
60. V.R. ÖLANDER: General list of astronomical azimuths observed in 1920-1959 in the primary triangulation net. Helsinki 1965. 47 pages. 1 map.
61. ERKKI KÄÄRIÄINEN: The second levelling of Finland in 1935-1955. Helsinki 1966. 313 pages. 1 map.
62. JORMA KORHONEN: Horizontal angles in the first order triangulation of Finland in 1920-1962. Helsinki 1966. 112 pages. 1 map.
63. ERKKI HYTÖNEN: Measuring of the refraction in the Second Levelling of Finland. Helsinki 1967. 18 pages.
64. JORMA KORHONEN: Coordinates of the stations in the first order triangulation of Finland. Helsinki 1967. 42 pages. 1 map.
65. Geodeettinen laitos - The Finnish Geodetic Institute 1918-1968. Helsinki 1969. 147 pages. 4 maps.
66. JUHANI KAKKURI: Errors in the reduction of photographic plates for the stellar triangulation. Helsinki 1969. 14 pages.
67. PENTTI KALAJA, V.R. ÖLANDER: Astronomical determinations of latitude and longitude in 1949-1958. Helsinki 1970. 242 pages. 1 map.
68. ERKKI KÄÄRIÄINEN: Astronomical determinations of latitude and longitude in 1954-1960. Helsinki 1970. 95 pages. 1 map.
69. AIMO KIVINIEMI: Niinisalo calibration base line. Helsinki 1970. 36 pages. 1 sketch appendix.
70. TEUVO PARM: Zero-corrections for tellurometers of the Finnish Geodetic Institute. Helsinki 1970. 18 pages.
71. ERKKI KÄÄRIÄINEN: Astronomical determinations of latitude and longitude in 1961-1966. Helsinki 1971. 102 pages. 1 map.
72. JUHANI KAKKURI: Plate reduction for the stellar triangulation. Helsinki 1971. 38 pages.
73. V.R. ÖLANDER: Reduction of astronomical latitudes and longitudes 1922-1948 into FK4 and CIO systems. Helsinki 1972. 40 pages.
74. JUHANI KAKKURI AND KALEVI KALLIOMÄKI: Photoelectric time micrometer. Helsinki 1972. 53 pages.
75. ERKKI HYTÖNEN: Absolute gravity measurement with long wire pendulum. Helsinki 1972. 142 pages.
76. JUHANI KAKKURI: Stellar triangulation with balloon-borne beacons. Helsinki 1973. 48 pages.
77. JUSSI KÄÄRIÄINEN: Beobachtungsergebnisse der finnischen Winkelmessungen in den Jahren 1969-70. Helsinki 1974. 40 S.
78. AIMO KIVINIEMI: High precision measurements for studying the secular variation in gravity in Finland. Helsinki 1974. 64 pages.
79. TEUVO PARM: High precision traverse of Finland. Helsinki 1976. 64 pages.
80. R.A. HIRVONEN: Precise computation of the precession. Helsinki 1976. 25 pages.
81. MATTI OLLIKAINEN: Astronomical determinations of latitude and longitude in 1972-1975. Helsinki 1977. 90 pages. 1 map.
82. JUHANI KAKKURI AND JUSSI KÄÄRIÄINEN: The Second Levelling of Finland for the Aland archipelago. Helsinki 1977. 55 pages.
83. MIKKO TAKALO: Suomen Toisen tarkkavaaituksen kiintopisteluettelo II. Bench mark list II of the Second Levelling of Finland. Helsinki 1977. 150 sivua.
84. MATTI OLLIKAINEN: Astronomical azimuth determinations on triangulation stations in 1962-1970. Helsinki 1977. 47 pages. 1 map.
85. MARKKU HEIKKINEN: On the tide-generating forces. Helsinki 1978. 150 pages.
86. PEKKA LEHMUSKOSKI AND JAAKKO MÄKINEN: Gravity measurements on the ice of Bothnian Bay. Helsinki 1978. 27 pages.
87. T.J. KUKKAMÄKI: Väisälä interference comparator. Helsinki 1978. 49 pages.
88. JUSSI KÄÄRIÄINEN: Observing the Earth Tides with a long water-tube tiltmeter. Helsinki 1979. 74 pages.
89. Publication dedicated to T.J. Kukkamäki on the occasion of his 70th anniversary. Helsinki 1979. 184 pages.
90. B. DUCARME AND J. KÄÄRIÄINEN: The Finnish Tidal Gravity Registrations in Fennoscandia. Helsinki 1980. 43 pages.
91. AIMO KIVINIEMI: Gravity measurements in 1961-1978 and the results of the gravity survey of Finland in 1945-1978. Helsinki 1980. 18 pages. 3 maps.
92. LIISI OTERMA: Programme de latitude du tube zénithal visuel de l'observatoire Turku-Tuorla systéme amélioré de 1976. Helsinki 1981. 18 pages.
93. JUHANI KAKKURI, AIMO KIVINIEMI AND RAIMO KONTTINEN: Contributions from the Finnish Geodetic Institute to the Tectonic Plate Motion Studies in the Area between the Pamirs and Tien-Shan Mountains. Helsinki 1981. 34 pages.
94. JUSSI KÄÄRIÄINEN: Measurement of the Ekeberg baseline with invar wires. Helsinki 1981. 17 pages.
95. MATTI OLLIKAINEN: Astronomical determinations of latitude and longitude in 1976-1980. Helsinki 1982. 90 pages. 1 map.
96. RAIMO KONTTINEN: Observation results. Angle measurements in 1977-1978. Helsinki 1982. 29 pages.

97. G.P. Arnautov, Ye N. Kalish, A. Kiviniemi, Yu F. Stus, V.G. Tarasiuk, S.N. Scheglov: Determination of absolute gravity values in Finland using laser ballistic gravimeter. Helsinki 1982. 18 pages.
98. Leena Mikkola (editor): Mean height map of Finland. Helsinki 1983. 3 pages. 1 map.
99. Mikko Takalo and Jaakko Mäkinen: The Second Levelling of Finland for Lapland. Helsinki 1983. 144 pages.
100. Jussi Kääriäinen: Baseline Measurements with invar wires in Finland 1958-1970. Helsinki 1984. 78 pages.
101. Raimo Konttinen: Plate motion studies in Central Asia. Helsinki 1985. 31 pages.
102. Raimo Konttinen: Observation results. Angle measurements in 1979-1983. Helsinki 1985. 30 pages.
103. J. Kakkuri, T.J. Kukkamäki, J.-J. Levallois et H. Moritz: Le 250^{e} anniversaire de la mesure de l'arc du meridien en Laponie. Helsinki 1986. 60 pages.
104. G. Asch, T. Jahr, G. Jentzsch, A. Kiviniemi and J. Kääriäinen: Measurements of Gravity Tides along the "Blue Road Geotraverse" in Fennoscandia. Helsinki 1987. 57 pages.
105. Jussi Kääriäinen, Raimo Konttinen, Lu Qiankun and Du Zong Yu: The Chang Yang Standard Baseline. Helsinki 1986. 36 pages.
106. E.W. Grafarend, H. Kremers, J. Kakkuri and M. Vermeer: Adjusting the SW Finland Triangular Network with the TAGNET 3-D operational geodesy software. Helsinki 1987. 60 pages.
107. Matti Ollikainen: Astronomical determinations of latitude and longitude in 1981-1983. Helsinki 1988. 37 pages.
108. Markku Poutanen: Observation results. Angle measurements in 1967-1973. Helsinki 1988. 35 pages.
109. Jussi Kääriäinen, Raimo Konttinen and Zsuzsanna Németh: The Gödöllö Standard Baseline. Helsinki 1988. 66 pages.
110. Jussi Kääriäinen and Hannu Ruotsalainen: Tilt measurements in the underground laboratory Lohja 2, Finland, in 1977-1987. Helsinki 1989. 37 pages.
111. Mikko Takalo: Lisäyksiä ja korjauksia Suomen tarkkavaaitusten linjastoon 1977-1989. Helsinki 1991. 98 sivua.
112. Raimo Konttinen: Observation results. Angle measurements in the Pudasjärvi loop in 1973-1976. Helsinki 1991. 42 pages.
113. Raimo Konttinen, Jorma Jokela and Li Quan: The remeasurement of the Chang Yang Standard Baseline. Helsinki 1991. 40 pages.
114. Jussi Kääriäinen, Raimo Konttinen and Markku Poutanen: Interference measurements of the Nummela Standard Baseline in 1977, 1983, 1984 and 1991. Helsinki 1992. 78 pages.
115. Juhani Kakkuri (editor): Geodesy and geophysics. Helsinki 1993. 200 pages.
116. Jaakko Mäkinen, Heikki Virtanen, Qiu Qi-xian and Gu Liang-rong: The Sino-Finnish absolute gravity campaign in 1990. Helsinki 1993. 49 pages.
117. Raimo Konttinen: Observation results. Geodimeter observations in 1971-72, 1974-80 and 1984-85. Helsinki 1994. 58 pages.
118. Raimo Konttinen: Observation results. Angle measurements in 1964-65, 1971, 1984 and 1986-87. Helsinki 1994. 67 pages.
119. Jorma Jokela: The 1993 adjustment of the Finnish First-Order Terrestrial Triangulation. Helsinki 1994. 137 pages.
120. Markku Poutanen (editor): Interference measurements of the Taoyuan Standard Baseline. Helsinki 1995. 35 pages.
121. Jorma Jokela: Interference measurements of the Chang Yang Standard Baseline in 1994. Kirkkonummi 1996. 32 pages.
122. Olli Jaakkola: Quality and automatic generalization of land cover data. Kirkkonummi 1996. 39 pages.
123. Matti Ollikainen: Determination of orthometric heights using GPS levelling. Kirkkonummi 1997. 143 pages.
124. Tiina Kilpeläinen: Multiple Representation and Generalization of Geo-Databases for Topographic Maps. Kirkkonummi 1997. 229 pages.
125. Jussi Kääriäinen and Jaakko Mäkinen: The 1979-1996 gravity survey and the results of the gravity survey of Finland 1945-1996. Kirkkonummi 1997. 24 pages. 1 map.
126. Zhitong Wang: Geoid and crustal structure in Fennoscandia. Kirkkonummi 1998. 118 pages.
127. Jorma Jokela and Markku Poutanen: The Väisälä baselines in Finland. Kirkkonummi 1998. 61 pages.
128. Markku Poutanen: Sea surface topography and vertical datums using space geodetic techniques. Kirkkonummi 2000. 158 pages
129. Matti Ollikainen, Hannu Koivula and Markku Poutanen: The Densification of the EUREF Network in Finland. Kirkkonummi 2000. 61 pages.
130. Jorma Jokela, Markku Poutanen, Zhao Jingzhan, Pei Weili, Hu Zhenyuan and Zhang Shengshu: The Chengdu Standard Baseline. Kirkkonummi 2000. 46 pages.
131. Jorma Jokela, Markku Poutanen, Zsuzsanna Németh and Gábor Virág: Remeasurement of the Gödöllö Standard Baseline. Kirkkonummi 2001. 37 pages.
132. Andres Rüdja: Geodetic Datums, Reference Systems and Geodetic Networks in Estonia. Kirkkonummi 2004. 311 pages.
133. Heikki Virtanen: Studies of Earth Dynamics with the Superconducting Gravimeter. Kirkkonummi 2006. 130 pages.
134. Juha Oksanen: Digital elevation model error in terrain analysis. Kirkkonummi 2006. 142 pages. 2 maps.
135. Matti Ollikainen: The EUVN-DA GPS campaign in Finland. Kirkkonummi 2006. 42 pages.
136. Annu-Maaria Nivala: Usability perspectives for the design of interactive maps. Kirkkonummi 2007. 157 pages.
137. Xiaowei Yu: Methods and techniques for forest change detection and growth estimation using airborne laser scanning data. Kirkkonummi 2007. 132 pages.
138. Lassi Lehto: Real-time content transformations in a WEB service-based delivery architecture for geographic information. Kirkkonummi 2007. 150 pages.
139. Pekka Lehmuskoski, Veikko Saaranen, Mikko Takalo and Paavo Rouhiainen: Suomen Kolmannen tarkkavaaituksen kiintopisteluettelo. Bench Mark List of the Third Levelling of Finland. Kirkkonummi 2008. 220 pages.
140. Eija Honkavaara: Calibrating digital photogrammetric airborne imaging systems using a test field. Kirkkonummi 2008. 139 pages.
141. Markku Poutanen, Eero Ahokas, Yuwei Chen, Juha Oksanen, Marita Portin, Sari Ruuhela, Heli Suurmäki (editors): Geodeettinen laitos – Geodetiska Institutet – Finnish Geodetic Institute 1918–2008. Kirkkonummi 2008. 173 pages.

142. Mika Karjalainen: Multidimensional SAR Satellite Images – a Mapping Perspective. Kirkkonummi 2010. 132 pages.
143. Maaria Nordman: Improving GPS time series for geodynamic studies. Kirkkonummi 2010. 116 pages.
144. Jorma Jokela and Pasi Häkli: Interference measurements of the Nummela Standard Baseline in 2005 and 2007. Kirkkonummi 2010. 85 pages.
145. Eetu Puttonen: Tree Species Classification with Multiple Source Remote Sensing Data. Kirkkonummi 2012. 162 pages.
146. Juha Suomalainen: Empirical Studies on Multiangular, Hyperspectral, and Polarimetric Reflectance of Natural Surfaces. Kirkkonummi 2012. 144 pages.
147. Leena Matikainen: Object-based interpretation methods for mapping built-up areas. Kirkkonummi 2012. 210 pages.
148. Lauri Markelin: Radiometric calibration, validation and correction of multispectral photogrammetric imagery. Kirkkonummi 2013. 160 pages.
149. Xinlian Liang: Feasibility of Terrestrial Laser Scanning for Plotwise Forest Inventories. Kirkkonummi 2013. 150 pages.
150. Eero Ahokas: Aspects of accuracy, scanning angle optimization, and intensity calibration related to nationwide laser scanning. Kirkkonummi 2013. 124 pages.
151. Laura Ruotsalainen: Vision-Aided Pedestrian Navigation for Challenging GNSS Environments. Kirkkonummi 2013. 180 pages.
152. Harri Kaartinen: Benchmarking of airborne laser scanning based feature extraction methods and mobile laser scanning system performance based on high-quality test fields. Kirkkonummi 2013. 346 pages.
153. Antero Kukko: Mobile Laser Scanning – System development, performance and applications. Kirkkonummi 2013. 247 pages.
154. Jorma Jokela: Length in Geodesy – On Metrological Traceability of a Geospatial Measurand. Kirkkonummi 2014. 240 pages.
155. Pyry Kettunen: Analysing landmarks in nature and elements of geospatial images to support wayfinding. Kirkkonummi 2014. 281 pages.
156. Mari Laakso: Improving Accessibility for Pedestrians with Geographic Information. Kirkkonummi 2014. 129 pages.

The name of the series has changed the 1st of January in 2015.

FGI Publications:

157. Lingli Zhu: A pipeline of 3D scene reconstruction from point clouds. Kirkkonummi 2015. 206 pages.
158. Robert E. Guinness: Context Awareness for Navigation Applications. Kirkkonummi 2015. 244 pages.

Finnish Geospatial Research Institute FGI
Vuorimiehentie 5
FI-02150 Espoo
Finland

www.fgi.fi